Using Shrinkage Methods for Model Selection and Improved Predictions:
An Application to Time to Degree for Transfer Students

By

MIRTA CASATI
THESIS

Submitted in partial satisfaction of the requirements for the degree of

MASTER OF SCIENCE

in

AGRICULTURAL AND RESOURCE ECONOMICS

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

James Chalfant, Chair

Jeffrey Williams

Dalia Ghanem

Committee in Charge

2022

Abstract

This work exploits machine learning (ML) techniques in a linear realm to select the best set of explanatory variables from a potentially large set. We dedicated particular attention to LASSO to explore how this technique improves a model's prediction accuracy. We also used an extension, Islasso, a method that allows hypothesis testing with parameters estimated with a penalized function. We explored the techniques using readily-available datasets and a novel dataset composed of 4,091 observations of UC Davis transfer students with 126 variables. We aimed at understanding UC Davis transfer students' performance, measured by time to degree. To highlight predictive differences, we divided the variables into two subsets: academic and personal. We concluded that academic variables are far more important for predicting students' time to degree. LASSO conducted on the academic subset resulted in the fewer misclassification errors and the lowest AIC, improving from an unpenalized model. Moreover, Islasso also showed that academic variables are the most important in predicting late graduation by transfer students. Lasso helped us understand which variables belonged in the model and reinforced many initial presumptions on which variables should have entered. Moreover, we showed that Islasso could be an excellent compromise to close the gap between inference and selection as it allows us to perform variable selection and to obtain reliable confidence intervals for a model's coefficients simultaneously.

Contents

1 Introduction

As described by Varian (2014), econometrics and data analysis is composed of 4 main categories:

- prediction;
- summarizing;
- estimation;
- hypothesis testing;

Machine learning, ML, is considered to be a sub-field of artificial intelligence (AI), and is focused on developing algorithms aimed at fitting possibly complex functions to provide a prediction of some variable as a function of a set of explanatory variables, at classifying data, and at clustering or grouping tasks (Varian 2014, Mullainathan and Spiess, 2017, Athey and Imbens, 2019).

We will not apply ML to look for the best fitting functional form for our data in this work. Instead, we will exploit ML in a linear realm to select the best set of explanatory variables from a potentially large set. The key behind ML is to follow an algorithm for model selection that improves upon other methods, for selecting purposes, such as stepwise regression or less systematic forms of data mining. The choice to limit consideration to models that are linear in the variables precludes a vast set of alternative functional forms, notably the ones with interactions between model variables. We will return to this point later.

In recent years, the prevalence of large datasets came at the cost of dealing with potentially more complex models and has exposed practitioners to new threats, such as higher risk of overfitting. Overfitting describes a scenario where the use of models with a large number of variables violates parsimony: "that is, that include more terms than necessary or use more complicated approaches than are necessary" (Hawkins, 2004). With many X variables to choose from, there is a strong possibility that researchers will include too many variables that fit well "in-sample", but that will not predict as well "out-of-sample". It is tempting to think that the best-fitting model in a given dataset will continue to perform best with new observations following the same data-generating process, but surprisingly, this is not necessarily the case. We will demonstrate this result in the applications that follow. While our results pertain directly to decisions about the variables to include in a model, we anticipate that similar conclusions apply to the selection of alternative functional forms.

Again, in this work, we will focus only on scenarios where overfitting occurs because variables deemed relevant using conventional measures, such as t-ratios, may turn out to be less helpful, if not completely irrelevant, when predicting out-of-sample, rather than working on functional forms. Overfitting results in models with a low bias in estimated coefficients but high variance: such models will perform well on a training dataset—a subset of the data used to train a model—but not as well on

the testing dataset—a subset of observations reserved to test the trained model. We shall remember that the results are highly sensitive to the random splitting of the sample into training and testing datasets: in case of overfitting the results will vary greatly for each training set, and therefore they will be poorly generalizable.

The relatively poorer performance of many models out-of-sample means that the model that fits the training sample best might not predict out-of-sample. For instance, a more parsimonious model estimated with the training dataset might produce better out-of-sample predictions if the bias that such a model introduces does less harm to a measure, such as MSE, than the reduction in variance that it brings.

We believe expanding traditional econometrics with ML methods could be a solution to face these new threats efficiently, enabling the practitioner to "sort out the mass of information and pare it down to its bare essentials" (Hastie et al; 2019). ML, therefore, helps the practitioner enforce the principle of parsimony, offering a solution via "regularization procedures," allowing the practitioner to obtain simpler models with good predictive power. Indeed, if there are many regressors p, the likelihood of overfitting increases making it advantageous to go one step further from least squares regression. We will relate the notion of "simple model" to the sparsity assumption; a sparse model is one where only a few predictors play an essential role (Hastie et al; 2019). As Hastie et al. (2019) explain, when $p \geq N$, there is an infinite set of solutions that make the objective function equal to zero, and "these solutions almost surely overfit the data as well" (Hastie et al; 2019). That is an extreme case, of course, nonetheless, when p is large, even if $p \leq N$, the risk of overfitting and obtaining poor out-of-sample predictions is still present. Therefore, we will apply linear shrinkage methods: Ridge, Lasso, and its extensions, to obtain sparse models.

First, we will theoretically introduce the linear shrinkage methods. Then, as mentioned above, we will dedicate particular attention to the extensions of traditional Lasso. Next, we will dedicate three sections to explore the shrinkage methods using two well-known and readily available datasets. As we will make extensive use of R and Stata in the first section, we will first try to match the two programs with the Mtcars dataset. Next, we will explore evaluating a model's quality of fit using the results from EAWE dataset. The following section will be dedicated to comparing two different cross-validation techniques to see which suits our purposes better.

Finally, the last chapter will apply linear shrinkage methods to a novel dataset composed of 4,091 observations of UC Davis transfer students. We aim to understand UC Davis transfer students' performance, measured by time to degree. As the dataset is made up of 126 variables, we believe that the use of shrinkage estimators might be beneficial to improve the model's prediction accuracy. Specifically, we will make use of Lasso regression to reach our purpose. Besides embracing the

problem from a prediction standpoint, we will explore the feasibility of a method to conduct inference on the parameters estimated with Lasso.

2 Shrinkage methods

Linear shrinkage methods constrain, or regularize, the coefficients in an estimated model, shrinking them towards zero or, in some cases, precisely to zero. The result is simpler models that are easier to generalize and interpret, aiming to balance "expressiveness against overfitting" (Athey, 2018).

When applying shrinkage methods, we need to keep in mind the bias-variance trade-off. Shrinkage methods impose more parsimony on the models, but this comes at a price, which is more bias in the estimator. Even though a simple model with few parameters might have more bias, parameter estimates may exhibit more efficiency, using for instance to lower squared error loss, thereby representing increased model prediction accuracy. Therefore, we must balance the cost of overfitting (high variance) against the cost of underfitting(high bias).

Consider a regression

$$y = f(\mathbf{x}) + u \quad \text{with} \quad E(u) = 0 \quad \text{and} \quad u \perp \mathbf{x}$$

The mean squared error of $\hat{\beta}_{ols}$ is:

$$\begin{aligned} MSE(\hat{\beta}) = E[(\hat{\beta} - \beta)(\hat{\beta} - \beta)'] = E[(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'u \cdot u'\mathbf{X}(\mathbf{X}'\mathbf{X}^{-1})] \\ = (\mathbf{X}'\mathbf{X})^{-\mathbf{1}}\mathbf{X}' \cdot E(uu') \cdot \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1} = \sigma^2(\mathbf{X}'\mathbf{X})^{-1} \end{aligned}$$

i.e. it is unbiased, therefore $MSE(\hat{\beta}) = cov(\hat{\beta}) + 0$.

For a biased estimator $\tilde{\beta}$:

$$\begin{aligned} MSE(\tilde{\beta}) = E[(\tilde{\beta} - \beta)(\tilde{\beta} - (\beta)'] = E[(\tilde{\beta} - E(\tilde{\beta}) + E(\tilde{\beta}) - \beta) \cdot (\tilde{\beta} - E(\tilde{\beta}) + E(\tilde{\beta}) - \beta)'] \\ = Cov(\tilde{\beta}) + Bias(\tilde{\beta})Bias(\tilde{\beta})' \end{aligned}$$

Therefore

$$MSE(a'\hat{\beta}) = a' \cdot \sigma^2(\mathbf{X}'\mathbf{X})^{-1}a \qquad \forall \quad a$$

$$MSE(a'\tilde{\beta}) = a' \cdot MSE(\tilde{\beta})a \qquad \forall \quad a$$

For instance, for $a = X_0$, these are the MSEs for predictions at $X_0'\hat{\beta}$ and $X_0'\tilde{\beta}$, respectively. It is possible for the reduction in the variance term MSE ($\tilde{\beta}$) to be great enough to offset the introduction of bias from using $\tilde{\beta}$. In any case, there is not a one-to-one correspondence between reducing the bias in the

parameter estimates and improving out-of-sample predictions of y. Shrinkage estimators minimize a criterion such as the RSS augmented by a penalty function. They enforce sparsity by adjusting estimated coefficients to take the penalty into account in the optimization function. Introducing this penalty term (often referred to as tuning parameter) in the objective function (i.e., regularization) generates a biased model with lower variance than a non-regularized model. For a given penalty function, the penalty term can be chosen by cross-validation, to minimize the MSE of prediction errors, or an alternative penalty may be used, such as AIC. In this work, we will estimate the penalty factor coefficient via cross-validation.

We dedicate the following sections to a brief analysis of each shrinkage method we exploit in our applications. We aim to highlight similarities and differences between estimators to clarify our choices when empirically applying them.

2.1 Ridge Regression

The first method we will discuss is Ridge regression, introduced by Hoerl and Kennard (1970), which may be preferable to OLS when there is multicollinearity among regressors and when estimators have large variances as a result. Multicollinearity occurs when two or more predictors are highly linearly correlated, and the correlation matrix for the estimated coefficients will be nearly singular. As Dougherty (2011) reports, "the higher the correlation between predictors, the larger the population variances of the distribution of their coefficients, the greater the risk of obtaining erratic estimates of the coefficients". As long as the other OLS assumptions are satisfied, for OLS, the estimated coefficients are still unbiased and consistent in the presence of multicollinearity, but their standard errors are larger; thus, the results might be less informative. The lack of precision in estimated coefficients means that it is difficult to distinguish between competing hypotheses about the parameter vector and out-of-sample estimates might be less reliable. Ridge regression is similar to least squares regression, other than the penalty term added to the objective function:

$$RSS = \sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2 + \lambda \sum_{j=1}^{p}\beta_j^2 = RSS + \lambda \sum_{j=1}^{p}\beta_j^2$$

Thus, the criterion to minimize is the usual RSS plus the penalty function $\lambda\sum_{j=1}^{p}\beta_j^2$. This is known in the ML literature as L_2 regularization. It shrinks the coefficients towards zero, but doesn't make them exactly equal to zero. λ is a tuning parameter which must be chosen or estimated separately. λ serves to balance the impact of RSS and the penalty term on the regression coefficient estimates. A different set of coefficients results from each λ. The shrinking penalty is applied from β_1 to β_p, and the intercept is not included.

The tuning parameter λ is always a positive value and can range from 0 to positive infinity, but typically is chosen to be between 0 and 10. As λ increases , the impact of the penalty grows and the elements of $\tilde{\beta}$, other than the intercept, will move closer to zero. When $\lambda = 0$, the penalty term has no effect, and $\beta_\lambda = (\tilde{\beta}_{ols})$ The larger is λ, the more these estimates differ, the smaller the covariance matrix for $\tilde{\beta}$, and the larger the bias in the Ridge estimator (Judge, 1988).

It is worth noting that Ridge parameters are not scale-invariant because if we re-scale the predictors, we obtain different coefficient estimates and predictions as the payoff from shrinking each individual coefficient is affected by unit of measurement in the X variables. In contrast, predictions from OLS are scale-invariant because if we re-scale the predictors, we still obtain the same predictions.

The estimated $\tilde{\beta}$ will depend on the units of the X variables, and a larger $\tilde{\beta}$ would contribute more to the penalty function.Thus, it is better to apply Ridge regression after standardizing the predictors (James et al; 2013).

How does this estimator solve the multicollinearity problem? Starting from

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{Y})$$

with the covariance matrix as follows:

$$Var\hat{\beta} = \sigma^2(\mathbf{X}^T\mathbf{X})^{-1},$$

suppose there is multicollinearity and $(\mathbf{X}^T\mathbf{X})$ is close to being singular. In that case, this means that even while it might still be invertible (as it is not exactly singular), the variance of the estimated coefficients will surely be large. As explained, Ridge minimizes RSS adding the penalty term $\lambda > 0$, thereby solving the "close to singularity issue." Adding a positive value of $\lambda > 0$ reduces the linearity effects between the columns, allowing us to estimate the model. As a result, along with a suitable choice of λ, the Ridge regression might outperform OLS in terms of MSE.

The ridge regression solution, denoted as $\tilde{\beta}$ is as follows:

$$\tilde{\beta} = (\mathbf{X}^T\mathbf{X} + \lambda I)^{-1}\mathbf{X}^T Y$$

Both simulations and empirical studies have evaluated Ridge Regression MSE. By simulation, Hoerl et al. (1975) showed that the mean squared error for Ridge regression coefficients is lower than OLS. Lawless and Wang (1976) and Dempster et al. (1977) also showed that Ridge regression has superior performance relative to OLS, a result which is dependent on picking a suitable λ.

On the other hand, by Monte Carlo simulation, McDonald and Galarneau (1975) proved that Ridge

does not always perform better than OLS in terms of MSE reduction. The authors highlight a specific issue that we will discuss later in our work. According to them, there is no constant value of λ which is guaranteed to yield an estimator better than least-squares in all cases. Moreover, they note that as there is no rule to choose $\lambda > 0$, it cannot be ensured that Ridge will always outperform OLS for any particular value of λ. In other words, there is no constant λ which improves Ridge upon OLS, but there is always some λ that improves Ridge upon OLS.

Shifting the focus to empirical applications, Ridge Regression has been applied to diverse research fields where practitioners exploited the tool to solve multicollinearity issues. For instance, Brown and Beattie (1975) showed the potential application of Ridge regression in economics. According to the authors, Ridge can be well suited for estimating the Cobb-Douglas production function. They provided an empirical example estimating the marginal productivity of irrigation water, where they show Ridge to be a better estimator than OLS, based on MSE comparison.

The authors concluded that in economics, we should specify a model entirely and "utilize available data and various prior information approaches for parameter estimation, rather than using unbiased estimation and mechanically deleting variables to reduce multicollinearity" For this approach, Ridge regression can be considered a "promising tool." Although Ridge history emphasizes multicollinearity over ML, it is very similar to techniques that are more directly the result of ML approaches. For this reason, we decided to include its description in this work: even if it doesn't perform variable selection, it is still shrinking coefficients aiming at reducing variance at the cost of increased bias, leading to potentially improved predictions.

2.2 Lasso Regression

An alternative to Ridge Regression is Lasso regression, which serves the same function of reducing model complexity, seeking to reduce the risk of overfitting. However, compared to Ridge, Lasso has an additional characteristic. As shown, Ridge shrinks the coefficients *towards* zero, but all p predictors will still be included in the final model. Instead, Lasso shrinks some of the parameters all the way to zero. As explained by Tibshirani (1996), Lasso is a regularization technique for **simultaneous** estimation and variable selection.

Let's think about the concept of sparsity, which we used as a proxy for simplicity; we understand why we need to take Ridge—the "great uncle of Lasso" (Efron and Hastie; 2016)—one step further. Lasso achieves this purpose by adding a different penalty to the loss function:

$$RSS = \sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2 + \lambda \sum_{j=1}^{p}\left|\beta_j\right| = RSS + \lambda \sum_{j=1}^{p}\left|\beta_j\right|$$

Therefore, the criterion to minimize is the usual RSS plus a penalty function $\lambda\sum_{j=1}^{p}|\beta_j|$. This is known as L_1 regularization, which, differently than Ridge, tends to shrink some of the coefficients precisely to zero. As a result, contrarily to Ridge, Lasso is likely to result in a model with only a subset of original regressors (James et al; 2013).

As for Ridge, λ selection is again of great importance. When $\lambda = 0$, we obtain the least-squares fit. When λ becomes sufficiently large, all the coefficients are set equal to zero. For values in between the extremes, some coefficients are restricted and others are not.

Following James et al. (2013), we will re-express the Ridge (1) and Lasso (2) equations in terms of a budget constraint, which will help understanding the difference embedded in the two shrinkage methods.

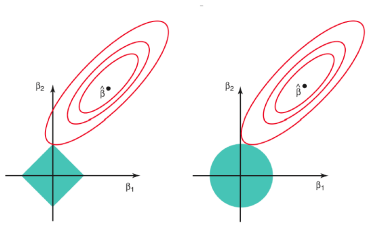
$$\min_{\beta} = \left\{ \sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p} -\beta_j x_{ij})^2 \right\} \quad \text{s.t.} \sum_{j=1}^{p} {\beta_j}^2 \leq s(\lambda) \tag{1}$$

$$\min_{\beta} = \left\{ \sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p} -\beta_j x_{ij})^2 \right\} \quad \text{s.t.} \sum_{j=1}^{p} \left|\beta_j\right| \leq s(\lambda) \tag{2}$$

In other words, as equation (2) shows, when performing Lasso, we are minimizing RSS, subject to the constraint that there is a budget s for how large $\sum_{j=1}^{p}|\beta_j|$ can be. The larger the s, the less restrictive the condition. If s is large enough, the solution obtained would be equal to the least squares solution. The same reasoning applies to the Ridge, with the difference that RSS is minimized subject to the restriction that $\sum_{j=1}^{p}\beta_j^2$ doesn't exceed s.

Exploiting the well known James et al. (2013) Lasso and Ridge 2-dimensional graphical representation ($p = 2$) (Figure 1), we observe that the Lasso constraint $\left|\beta_1\right| + \left|\beta_2\right| \leq s$ has a diamond shape, whereas the Ridge constraint is a circle ${\beta_1}^2 + {\beta_2}^2 \leq s$. The ellipsis centered around $\hat{\beta}$ represents constant values of RSS, meaning that all the points on a given ellipsis represent the same RSS. The constraint implied by Ridge regression is a circle. It will not generally have any corner solutions at the axes, so interior solutions will almost always occur. In contrast, the Lasso diamond-shaped constraint seems more likely to result in corner solutions, as shown in Figure 1, where $\beta_1 = 0$, so the resulting model will only include $\beta_2 = 0$.

Figure 1



Under this Lagrangian formulation, the similarities between Lasso, Ridge, and another classical method for selection and estimation in a linear model, best-subset selection (Beale et al; 1967, Hocking and Leslie, 1967) become evident. Best-subset selection aims at finding a small subset of predictors to minimize RSS. The best selection approach is appropriate when the number of all possible predictors is small because it is otherwise too computationally expensive. The algorithm considers all of the 2^p combinations of the available regressors, including the null model, as possible for selection. James et al. (2013) expressed the best selection as follows:

$$\min_{\beta} = \left\{ \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} -\beta_j x_{ij})^2 \right\} \quad \text{s.t.} \sum_{j=1}^{p} I(\beta_j \neq 0) \leq s(\lambda) \tag{3}$$

where $I(\beta_j \neq 0)$ is an indicator variable, taking on value of 1 if $(\beta_j \neq 0)$, and equals zero otherwise. In this case, RSS is minimized subject to the constraint that no more than s coefficients can be nonzero.

It becomes clear that when p is large, it can be computationally unfeasible or hard, at the very least, to use the best selection approach. Best selection considers all the possible $\binom{p}{s}$ models containing s predictors. Therefore Lasso and Ridge can be considered computationally feasible alternatives to best selection with easier constraints to solve.

2.3 Extensions of Lasso

As discussed in the previous section, Lasso will enforce an L_1 penalty, which brings a sparsity payoff but also some drawbacks. The shrinkage makes the objective function non-smooth, resulting in parameter estimates that are biased towards zero and complexities for the standard error computation (Ciluffo et al; 2020).

The literature has discussed different extensions to Lasso to remedy these drawbacks. Specifically, in this work we will focus on Adaptive Lasso (Zou, 2006), Relaxed Lasso (Meinshausen,2007), and induced smoothed Lasso (Ciluffo et al; 2020).

2.3.1 Adaptive Lasso

Zou (2006) introduced adaptive Lasso, a two-stage approach which extends traditional Lasso where "adaptive weights are used for penalizing different coefficients in the L_1 penalty ". The Lasso criterion is modified by introducing a vector of weights $\hat{w}$ in the penalty function:

$$RSS = \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij})^2 + \lambda \sum_{j=1}^{p} \hat{\mathbf{w}}_\mathbf{j} \left|\beta_j\right| = RSS + \lambda \sum_{j=1}^{p} \hat{\mathbf{w}}_\mathbf{j} \left|\beta_j\right|$$

where $\hat{\mathbf{w}}$ is the weight vector $\hat{\mathbf{w}}_\mathbf{j} = \frac{1}{|\hat{\beta}_j^{\gamma}|}$. Estimates from OLS or Ridge can be used to calculate the weights.

The author points out that traditional Lasso doesn't have "oracle properties", meaning that it is not consistent for parameter estimation and variable selection. Instead, he points out that if the weight vector is "data-dependent and cleverly chosen" (Zou, 2006), Lasso can be consistent for selection and estimation. To compute the weights, we need to calculate some initial estimates, preferably consistent and unbiased.

For this reason, Zou suggests the application of OLS, but other consistent estimators can also be used. Furthermore, as the author specifies, if collinearity is a concern, Ridge regression can also be used to estimate the weights, even though it produces estimates that are biased and inconsistent. In that case, it is not guaranteed that the Oracle property will hold.

After selecting the $\hat{\beta}$ values to use for weights, we would need to find an optimal pair of γ and λ. To do so, we can either use two-dimensional cross-validation or exclusively search for the optimal λ, as the value of γ can also be heuristically chosen from values such 0.1, 0.5, 1, 2 as Chen et al. (2019) showed. Once we find the optimal γ and λ pair, we would use the previously estimated coefficients to define the weights. If the estimated coefficients are large then they correspond to a smaller weight, and thereby less effect on the penalty function in the penalized Lasso estimation.

Finally, we perform Lasso with the modified penalty factor specification, which leads, according to Zou, to consistency in both selection and estimates.

Zou shows by simulation that adaptive Lasso performs better when compared to other "sparse modeling techniques". Huang et al. (2006) studied adaptive LASSO in high-dimensional settings. The authors showed that even when the performance of adaptive Lasso and traditional Lasso are similar, the number of variables selected by the former method is lower than the one chosen by the latter, and the prediction MSE of adaptive Lasso is smaller in this sense, it is guaranteed to improve upon traditional Lasso.

In our opinion, the adaptive Lasso strategy is pretty arbitrary. Zou affirms that we can use either OLS or Ridge estimates as adaptive weights but that if we choose Ridge, there is no guarantee that the Oracle property will hold. As to our knowledge, the adaptive Lasso strategy isn't fully clear; we decided to use other extensions to Lasso in the following empirical applications.

2.3.2 Induced Smoothed Lasso

Ciluffo et al. (2020) propose an extension to traditional Lasso that aims to get reliable p-values and thus permit hypothesis testing. As the authors explain, the L_1 regularization makes the objective function non-smooth. Therefore there might be "limitations" in the computation of standard errors (Ciluffo et

al; 2020).

Following Brown and Wang (2005), Ciluffo et al. (2020) apply the induced smoothing approach to the Lasso objective function. The Brown and Wang (2005) induced smoothing method approximates discontinuous but monotone estimating functions using continuously differentiable functions.

The Islasso idea is to consider the new estimating equation obtained by averaging the non-smooth score over scaled normal perturbations of the parameters. Smoothing the traditional Lasso loss function implies that the estimates from Islasso will never be precisely zero, although the values obtained can be very small.

The smoothing process allows obtaining Wald statistics. In the usual setting for inference, a single hypothesis $H_0 : \beta = 0$ versus $H_1 : \beta \neq 0$ can be tested using the Wald statistics:

$$W = \frac{\hat{\beta}}{SE(\hat{\beta})}$$

For simpler cases, such as OLS, the Wald test follows a standard normal distribution; unfortunately, the distribution is unknown for traditional Lasso. Islasso solves this issue because its sampling distribution has "no probability mass but a smoothed peak around zero", closer to a Normal distribution.

To practically implement Islasso Ciluffo et al. (2020) developed the `islasso` package in R, which follows the heuristic framework previously explained, thereby returning reliable standard errors and p-values, which we can use to perform inference on the coefficients.

2.3.3 Relaxed Lasso

Meinshausen (2007) developed a two-stage procedure defined as Relaxed Lasso to overcome the slow convergence of traditional Lasso when there are many variables in a dataset. He points out that it may not be optimal to control shrinkage estimation and model selection with a single λ parameter. To solve this concern, Meinshausen introduced a second parameter ϕ, which acts as a multiplicative factor of λ controlling for the shrinkage applied to the subset of variables included in the model for a specific λ value. If $\phi = 1$, then Lasso and Relaxed Lasso coincide, whereas if $\phi = 0$ the solution is OLS for the subset of variables selected with λ.

The relaxed Lasso is defined for $\lambda \in [0, \infty)$ and $\phi \in (0, 1]$. Following Meinshausen:

$$\hat{\beta}^{\lambda,\phi} = argmin \quad n^{-1} \sum_{i=1}^{n} (Y_i - X_i^T \{\beta \cdot 1_{M_\lambda}\})^2 + \phi\lambda \left\|\beta\right\|_1$$

In general, the Relaxed Lasso produces sparser results compared to Lasso. To find the relaxed Lasso solutions the first step is to find the Lasso solutions for all λ, and then for each λ, refit the

variables in the active set without any penalization (Hastie et al; 2021).

As Meinshausen (2007) shows, Lasso and Relaxed Lasso produce similar results, but when the number of relevant predictor variables is large, there is a relative improvement in using Relaxed Lasso. Moreover, the Relaxed Lasso solutions have no extra computation cost but lead to sparser models and more accurate predictions (Meinshausen, 2007).

3 Tuning parameter selection

Due to its critical importance, we decided to dedicate a section to explain how we approached the problem of selecting the tuning parameter λ. As described above, the tuning parameter has the aim to determine the shrinkage strength. As λ increases, the impact of the shrinkage penalty grows. We decided to apply cross-validation (CV) to select λ. To fully understand how CV works, it might be helpful to take a step back to the more traditional validation approach.

To perform the validation approach, we split the original sample of size n into two parts: a training set and a validation set. First, we estimate the model using the training set, and the fitted model is used to predict the responses for the observations in the validation set. Then the out-of-sample mean squared error for the test data is computed as follows.

$$\sum_{i=1}^{n_{test}} (y_i - \hat{y}_i)^2 / n_{test}$$

This approach lets us compare the out-of-sample predictive power of various competing models. Model selection can then be based on the model that performs best at predicting the predicting the observations in the validation set. A key point is that the best-fitting model "in sample" (i.e. fit using the training set) may not perform best out-of-sample. The validation approach has drawbacks. First, as the chosen sample split is arbitrary, the calculation of the test error rate is variable because it depends on which observations are included in the training and the validation sets: different researchers will, in general, obtain different results. Secondly, only the observations in the training set (in general half of the original set) are used to fit the model and since models tend to perform worse when trained on fewer observations "there is the risk that the validation set could overestimate the test error rate for the model fit on the entire dataset" (James et al; 2013).

Cross-validation—a refinement of this traditional validation approach—is a re-sampling method that uses different portions of a dataset (folds) to measure the out-of-sample predictive of a model. There are different ways to perform CV. In this work, we applied Leave-one-out cross-validation (LOOCV), a particular case of CV where the number of folds equals the sample size. This is also

known as the jackknife.

When we perform LOOCV, a single observation is used for the validation set and the remaining observations form the training set. We fit the model on the $n-1$ training observations: we remove one observation i (test data) and estimate the rest of the sample (training data). Then, we compute $\hat{Y}_i$ for the excluded observation (the test observation) and the prediction error. The prediction error is an approximately unbiased estimate for the test error because $observation_i$, was not used in the fitting process: $\hat{e}_i* = y_i - \hat{y}_i$.

We repeat the procedure by selecting each observation for the validation data and then train the process on the other $n-1$ observations. Repeating the procedure produces n squared errors, and we average these n to obtain

$$\sum_{i=1}^{n}(y_i - \hat{y}_i)^2/(n)$$

LOOCV allowed us to have no randomness in the data splits present in other cross-validation forms. In the validation approach and other CV forms, such as $k-folds$ cv, we will obtain different results due to the randomness in the training/validation splits. Instead, performing LOOCV will always produce results that can be replicated by others.

One of the main drawbacks of LOOCV is that it can be computationally expensive when the dataset is of large dimensions, as we have to carry out the procedure for each observation in the dataset.

To choose λ, we will compute the cross-validation error for each value of λ and then select the λ value for which the cross-validation error is the smallest. Once λ is chosen, we fit the shrinking procedure with the selected value of the tuning parameter. Performing LOOCV to select λ allows us to obtain a unique λ value, which provides replicability and more consistency in our results.

4 Mtcars dataset: matching Stata and R

This work will extensively use both R and STATA; both programs have built-in functions for ML. Even experienced users might find it difficult to understand these programs, especially in trying to reconcile the results. Thus, the first application section will be dedicated to exploring how R and Stata work, with a familiar, readily-available dataset. Our goal is to open the "black box," namely to understand how the various commands work and under which conditions they give the same results. The interpretation of results from ML commands is complicated by the fact that various commands seem to give different results when, in fact, they are equivalent, as we shall see.

Our primary aim will be to obtain matching results between R and Stata. For this purpose, we will use the R built-in Mtcars dataset to predict miles per gallon based on a car's other characteristics. We will follow this online example: https://www.datacamp.com/community/tutorials/tutor

ial-ridge-lasso-elastic-net, to ensure that we can replicate the example results in R and then duplicate them in STATA. The result should be both deeper understanding of the methods and added confidence in our implementation of ML to new datasets that follow.

The Mtcars data were extracted from the US magazine Motor Trend in 1974 and comprised 11 explanatory variables that describe characteristics and performance for 32 automobiles. Our aim is to explore the relationship between this set of variables and miles per gallon (MPG). In Table 1, we report variable definitions taken from https://rpubs.com/neros/61800:

Table 1

#	Var	Description	Comments
0	name	Model of Vehicle	
1	mpg	Miles/US Gallon	
2	cyl	Number of cylinders	
3	disp	Displacement	This metric gives a good proxy for the total amount of power the engine can generate.
4	hp	Gross horsepower	Gross horsepower measures the theoretical output of an engine's power output.
5	drat	Rear axle ratio	The rear axle gear ratio indicates the number of turns of the drive shaft for every one rotation of the wheel axle.
6	wt	Weight (lb/1000)	
7	qsec	1/4 mile time	A performance measure, primarily of acceleration.
8	vs	V/S	Binary variable signaling the engine cylinder configuration.
9	am	Transmission Type	A binary variable signaling whether vehicle has automatic or manual transmission configuration.
10	gear	Number of forward gears	Number of gears in the transmission.
11	carb	Number of carburetors	The number of carburetor barrels.

We will begin with OLS and Ridge Regression estimates, both in R and Stata, and continue with the same approach for Lasso Regression. The advantage of starting with OLS is that OLS estimates would be easily replicable across programs, and by other users, and it also lets us demonstrate the effects of shrinking of the coefficients, by providing a model for comparison. Our final aim is to obtain a successful match between the two programs. In addition, we will dedicate a section to compare LOOCV and k-folds CV, to justify using the former approach.

4.1 Mtcars Ridge Regression in R and Stata

To replicate this https://www.datacamp.com/community/tutorials/tutorial-ridge-lasso-elastic-net example in R we would need to use the `glmnet` package. We started by preparing the data. First, we set $y = mpg$, and then we created the matrix of explanatory variables, $\mathbf{X}$, containing the 11 variables from Mtcars. We centered y, so that $y_i = mpg_i - \overline{mpg}$ whereas $\mathbf{X}$ will be standardized when fitting the model. Centering y and standardizing $\mathbf{X}$ are essential procedures to carry out Ridge Regression correctly. As mentioned earlier, Ridge regression is not scale-invariant. Therefore if the explanatory variables are not standardized, the contribution of a variable's coefficient β_i to the penalty function $\lambda \sum_{j=1}^{p} \beta_j^2$ will depend on units of measurement. In natural units of measurement, the largest β_i is the best candidate for shrinking. It is more desirable to express all x_i variables using comparable units of measurement.

After the re-scaling, we searched over values for λ by using the function `cv.glmnet`, which is

the main function to perform cross-validation in R. We specified `nfolds=32` to conduct LOOCV. We decided to perform LOOCV because, as previously explained, LOOCV allows us to have no randomness in the data splits, thereby obtaining a unique λ value, which provides replicability of results. In comparing the results between different programs, the results otherwise might differ solely due to different outcomes from random optimization. Moreover, we set $\alpha = 0$ to specify that we wanted to conduct Ridge Regression. It's worth emphasizing that we conducted `cv.glmnet` specifying the option `standardize = TRUE`. The package `glmnet` performs standardization but then reports the coefficients unstandardized. From the object cv.glmnet returned we selected, by cross-validation, the value of `lambda.min`, which is the value of λ that minimizes the MSE. Another option would be to compare the values for some model-selection criterion, such as Akaike's Information Criterion (AIC),

$$AIC = nlog\frac{azsum^{n}_{iSA=1}(Y_i - \hat{Y}_i)^2}{n} + 2(p+2)$$

or the Bayesian Information Criterion (BIC),

$$log\frac{\sum^{n}_{i=1}(Y_i - \hat{Y}_i)^2}{n} + log(n)(p+2)$$

which differently than AIC also imposes a penalty on N: as N increases the penalty would be higher.

Before fitting the Ridge model, we standardized **y** as well to obtain results scaled in the same way. It's worth noticing that `glmnet` fit the Ridge regression for 100 λ values. As we are interested in the value of λ which minimizes MSE, we will fit Ridge regression specifying that the coefficient on the penalty function is the λ_{min} selected via CV. Finally, from the `glmnet` object we extracted the estimated coefficients. Up to this point, we replicated the https://www.datacamp.com/community/tutorials/tutorial-ridge-lasso-elastic-net online example to make sure to gain confidence with `glmnet` and how the algorithm works in R. We decided to take a step further by experimenting with another function in R, `lmridge`. Our aim is to check whether `lmridge` and `glmnet` agreed on the coefficients estimated.

The `lmridge` function fits linear ridge regression after scaling the regressors and centering the response. To fit lmridge we used the previously chosen λ_{min} by `cv.glmnet`.

As Table 2 shows, we obtained similar results between `glmnet` and `lmridge` by using the option `scaled` for the latter command. In the `scaled` option centered values for the predictors (other than the intercept) are divided by the sample standard deviation of equal predictors. `lmridge`. In the table, we also report OLS results to show the effect of using Ridge to shrink the coefficients.

Having understood how R works we will then move to Stata, using `ridgereg` to perform Ridge regression. To see whether `ridgereg` matched our results from R, we performed Ridge Regres-

Table 2

Ridge Regression	**OLS**	**Glmnet**	**lmridge**
Intercept	12.3034	-0.0980	-0.0949
cyl	-0.1114	-0.0421	-0.4237
disp	0.0133	-0.0003	-0.0033
hp	-0.0215	-0.0022	-0.0022
drat	0.7871	0.1627	0.1621
wt	-3.7153	-0.3129	-0.3119
qsec	0.8210	0.0516	0.0514
vs	0.3178	0.0793	0.0804
am	2.5202	0.3496	0.3496
gear	0.6554	0.1045	0.1048
carb	-0.1994	-0.1098	-0.1101

sion setting λ_{min}=2.746789, as previously estimated with LOOCV in R. Moreover, we specified the `option=orr`, to perform ordinary ridge regression. We matched the results between `ridgereg` and `lmridge` specifying the scaling option `sc` in the latter command (Table 3). In the scaling option `sc` the divisor is $\sqrt{\sum_{j=1}^{p}(x_i - \bar{x}^2)}$ i.e. each variable in the scaled case is multiplied by $\frac{1}{\sqrt{n-1}}$. We report the results in Table 3. Hence R's `lmridge` and Stata's `ridgereg` give identical results, but only if we change from the `scaled` option to `sc`.

Table 3

Ridge Regression	**Ridgereg**	**lmridge**
Intercept	-0.65925	-0.65925
cyl	-0.32125	-0.32125
disp	-0.00465	-0.00465
hp	-0.00818	-0.00818
drat	0.86858	0.86858
wt	-0.70309	-0.70309
qsec	0.14058	0.14058
vs	0.81302	0.81302
am	0.95708	0.95708
gear	0.43880	0.43880
carb	-0.28942	-0.28942

As shown above we have two options to perform Ridge regression in R: `lmridge` and `glmnet`. We managed to reconcile Stata's `ridgereg` and `lmridge` easily by just specifying `sc` as scaling option. Reconciling R's `glmnet` with Stata's `ridgereg` wasn't as straightforward.

`glmnet` starts from a modified objective function, which is divided by the number of observations and has standardized response variables (Y is divided by its standard deviation). The result is that we obtain

$$\widehat{\beta_{glmnet}} = (\mathbf{X}^T\mathbf{X} + N\lambda I)^{-1}\frac{\mathbf{X}^T Y}{SD_Y}$$

To retrieve the original Ridge solution, we would need therefore to scale λ by a factor equal to $\frac{SD_Y}{N}$.

Finally, we decided to compute the coefficients for Ridge Regression without making the use of

any package, but simply calculating

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X} + \lambda I)^{-1}\mathbf{X}^T Y$$

with manually standardized coefficients as follows

$$\frac{X - \bar{X}}{\sqrt{c * var(X)}}$$

where $c = n - 1$, for `sc` and $c = 1$ for `scaled`.

```
sd_y <- sqrt(var(y)*(n-1)/n)
pmatrix=diag(p)
pmatrix[1,1]=0
beta1 <- solve(t(x1)%*%x1+lambda*(pmatrix),t(x1)%*%(y))
```

Once obtained this results we estimated Ridge regression with `glmnet`. Along with rescaling the coefficients, before computing the solution we also we would need therefore to scale λ by a factor equal to $\frac{SD_Y}{N}$. We proceeded as follows where where $N = 32$ and $p = 11$.

```
scale=(31/32)*sqrt(var(y))
ridge_cv <- cv.glmnet(Xs, y, alpha = 0, centered=FALSE,
standardize = T, grouped=FALSE, nfolds = 32)
lambda=ridge_cv$lambda.min
ridgematch=glmnet(Xs,y, alpha=0, standardize = F,lambda=scale*lambda/32)
```

Next, we estimated Ridge regression with `lmridge`. To match the previous results we used the option `sclaling=sc`.

```
y <- mtcars %>% select(mpg) %>%
scale(center = TRUE, scale = FALSE) %>% as.matrix()
lmridge=lmridge(y~Xs1,data=mtcars,scaling="sc",K=lambda)
```

At this point, we observed that `option=sc` is the option we used to match Stata's `ridgereg` and `lmridge`. Therefore we proceeded with the manual scaling of the coefficients in Stata before fitting `ridgereg`. As Table 4. shows, this procedure allowed us to close the loop and finally obtain closely matching results across `ridgereg`, `glmnet` and `lmridge`.

Table 4

X	$\hat{\beta} = (\mathbf{X}^T\mathbf{X} + \lambda I)^{-1}\mathbf{X}^T Y$	Glmnet (R)	Lmridge (R)	Ridgereg (Stata)
cyl	-3,1943	-3,2079	-3,1943	-3,1943
disp	-3,2089	-3,2234	-3,2089	-3,2089
hp	-3,1226	-3,1398	-3,1226	-3,1226
drat	2,5857	2,5978	2,5858	2,5857
wt	-3,8303	-3,8558	-3,8303	-3,8303
qsec	1,3987	1,4032	1,3987	1,3987
vs	2,2815	2,2889	2,2815	2,2815
am	2,6590	2,6763	2,6590	2,6590
gear	1,8026	1,8103	1,8026	1,8026
carb	-2,6027	-2,6219	-2,6028	-2,6027

4.2 Mtcars Lasso Regression in R and Stata

We observe that the matching between R and Stata of Lasso Regression is straightforward than Ridge Regression. First, we performed Lasso Regression in R. To do so, we used the same package as for Ridge regression, the `glmnet` package.

As with Ridge Regression, we centered y, whereas $\mathbf{X}$ will be standardized by `glmnet` when fitting the model. As Ridge, Lasso isn't scale-invariant: the size of the constraint put on each coefficient depends on the units of the X variables. For this reason, we standardized the variables before fitting Lasso.

We performed cross validation using `cv.glmnet`. Again, we specified `nfolds=32` to conduct LOOCV. Moreover, we set $\alpha = 1$ to specify that we wanted to conduct Lasso Regression, as we did for Ridge Regression, we specified the option `standardize=TRUE`, to perform cross validation: `cv.glmnet(X,y,alpha=1,standardize=TRUE, nfolds =32,grouped=FALSE)`, the option `grouped=FALSE`, is enforced by R when carrying out LOOCV. `glmnet` fits the model for 100 values of λ. As our aim is to obtain information on the predicted values estimated with λ_{min}, value of λ which minimizes MSE, from the cv.glmnet object we extracted λ_{min} and re-fit Lasso Regression with this λ value as penalty.

Using λ_{min} we estimated Lasso: `(lasso.mod=glmnet(X,y,alpha=1,lambda=lambda.min)`. From the `glmnet` object we extracted the estimated coefficients.

As previously mentioned, the matching with Stata was easy to obtain. First, we centered y. Then, we decided to perform cross-validation with `elasticnet` as we did for Ridge Regression. For Lasso, it is not necessary to conduct cross-validation using `elasticnet` as Stata conducts cross-validation and Lasso within the same command `Lasso linear`.

In fact, our purpose was not to conduct CV per se, but rather to see whether `elasticnet` agreed with `cv.glmnet`. We found out that, contrary to Ridge, we successfully obtained the same λ_{min} as in R [0.7295713].

We then fit Lasso regression, `lasso linear centered_mpg X,selection(cv)folds(32)`. To obtain the coefficients in the same scale as R, we needed to specify the option `penalized`.

As Table 5 shows, we successfully matched Stata and R results. As we observe, differently than Ridge, some coefficients were shrunk exactly to zero. The shrinkage tells us to include three explanatory variables in the model: the number of cylinders (cly), gross horsepower (hp), and overall weight of the vehicle (wt), which according to Lasso, will predict the miles per gallon consumed by the cars.

Table 5

Lasso	**R**	**Stata**
Intercept	16.1534	16.15333
cyl	-0.8907	-0.88947
hp	-0.0123	-0.01228
wt	-2.7492	-2.74963

5 EAWE dataset: learning how to evaluate a model quality of fit

To understand how well shrinking methods perform on a given dataset, we need to measure how the predictions match the observed data. Therefore, the EAWE application will be used to examine how to evaluate a model quality of fit.

EAWE stands for Educational Attainment, and Wage Equations dataset which, as mentioned above, which is used by Dougherty in his book "Introduction to Econometrics". The dataset is used to estimate various models of the determinants of earnings and educational attainment. The dataset is actually a subset of the National Longitudinal Survey of Youth, a panel survey with repeated interviews of a sample of young males and females aged 12 to 18 in 1997. This dataset is supplied as 22 subsets, each consisting of 500 observations, 250 drawn from male respondents and 250 from female respondents. Each of these 22 datasets contains the same 97 variables. We combined these 22 datasets into a merged EAWE dataset containing 11,000 observations.

The subsets' random structure allowed us to conduct a train-test evaluation of the estimators efficiently. In the following applications we will use the first 1000 observations; i.e. the datasets EAWE01 and EAWE02.

We discussed the importance of measuring out-of-sample predictive ability using cross-validation, used in an ML setting to evaluate a model's goodness-of-fit. The goodness of fit is a very familiar concept. In OLS, the R^2 is the goodness-of-fit measure that is typically reported. However, the $\hat{y}_i$ values that are used to calculate the RSS, and consequently the R^2 are in-sample predictions which, as already noticed, are subject to "overfitting". This means that the R^2 might give an overly optimistic

picture of a model's predictive power for observations beyond the sample used for estimation.

To measure how well the prediction matches the observed data, we will use the mean squared error, MSE. If the MSE is small, then the predicted responses are close to the actual reactions. Instead, if it's large, the prediction and the true responses differ by a lot. The MSE, however, is calculated based on "out-of-sample" predictions.

As before, model selection begins with the choice of the tuning parameter λ. We aim to find the value of λ that corresponds to the lowest test MSE, i.e. out-of-sample MSE, not the lowest training MSE, calculated using the training sample. The most common procedure is to divide the data into training and test; the usual ratio is 80%(training)-20%(testing).

Having picked a value for the training parameter, we will use only the training observations to estimate the various model we consider. In our case, we will fit Ridge and Lasso only on training data, and then we will obtain predictions using the testing data. Once the predictions are obtained, we will evaluate the test MSE:

$$MSE = \sum_{i=1}^{n_{test}} (y_{test} - \widehat{y_{test}})^2/(n_{test})$$

5.1 EAWE Ridge Regression Quality of Fit

As $\hat{\beta}_{ols}$ has the smallest MSE among all linear unbiased estimators, assuming the model is correctly specified, we deemed it self-evident to compare MSE_{ridge} to MSE_{ols}, to see whether Ridge Regression, which increases the bias and reduces the variance, would perform better than OLS.

For this purpose we decided to work with a set of predictors exhibiting multicollinearity. Specifically, following Dougherty's (2011) example, we will use highly correlated ASVABC variables:

- ASVABC, a composite measure of cognitive ability, constructed with the scores of tests of arithmetic reasoning,
- ASVABC4, paragraph comprehension
- ASVABAR, arithmetic reasoning;
- ASVABWK, word knowledge

We specified a model and EARNINGS as the response variable and S (years of schooling), ASVABC, ASVABC4, ASVABAR, ASVABWK as the independent variables. In Table 6 we report the correlation matrix for the variables used in the regression. As shown, the "ASVABC" variables are highly correlated.

As mentioned, the EAWE structure offered us the possibility to conduct training and test evaluations without splitting the data. However, as the dataset comprises 22 subsets, each consisting of 500 random observations, we selected two subsets, EAWE01 and EAWE02, corresponding to 1000

Table 6

	ASVABC	ASVABC4	ASVABAR	ASVABMK	S
ASVABC	1				
ASVABC4	0.9841	1			
ASVABAR	0.9389	0.8986	1		
ASVABMK	0.8184	0.8919	0.7945	1	
S	0.5309	0.5618	0.4706	0.5463	1

observations. Therefore, we did not divide the observation training and testing set using random sampling: we used 500 observations from EAWE01 and 300 from EAWE02 to make the training set and the remaining 200 from EAWE02 for the testing set. This corresponds to the common 80% and 20% split.

First, we estimated OLS on the training set, and we computed MSE_{ols}. The MSE we obtained is 0.3273358. Next, we computed Ridge Regression. The steps we followed are structurally identical to the ones described for the Mtcars application. We fit Ridge Regression using training data (800 observations), then we extract the predicted values using testing data (the remaining 200 observations). First, we estimated the λ penalty value. To do so, we conducted LOOCV on the training set. From the LOOCV, we extracted λ_{min} and we fit Ridge Regression on the training dataset using this value of λ. Finally, we extracted the prediction on the last 200 observations.

```
ridge_cv <- cv.glmnet(x[1:800], y[1:800], alpha=0,
standardize=TRUE, nfolds=800, grouped=FALSE)
ridge.mod=glmnet(x1,y[1:800],alpha=0,lambda=lambdamin)
ridge.pred=predict(ridge.mod,s=lambdamin,newx=x[801:1000])
```

We computed the test MSE for Ridge Regression by subtracting the fitted values on the testing set from the testing observed y: `mean((ridge.pred-y.test)^2`. We obtained 0.250398, which shows an improvement compared to the 0.3273358 OLS MSE, meaning that Ridge Regression offers better predictions for the 200 observations in the testing dataset.

5.2 EAWE Lasso Regression Quality of Fit

As we did for Ridge Regression, we compared MSE_{lasso} to MSE_{ols}, to see whether Lasso Regression offers better performance than traditional OLS. We decided to fit OLS and Lasso on the entire set of EAWE variables, to strain Lasso's variable selection ability. Once we removed the response variable (EARNINGS) from the dataset and the variable ID, the dataset contains 94 variables.

First, we fit OLS on the training data and we computed MSE_{ols}. As a result, we obtained MSE_{ols}= 0.4205839. Then, we proceeded with LOOCV, needed to select the optimal λ penalty value to fit Lasso Regression. From the `lasso_cv` object we extracted λ_{min} and we fit a Lasso regression with $\lambda = \lambda_{min}$

on the training set. Finally, we extracted the predicted values on the testing data.

```
lasso_cv <- cv.glmnet(newdata[1:800,], y[1:800], alpha = 1,
standardize = TRUE, nfolds = 800, grouped=FALSE)
lasso.mod=glmnet(x[1:800],y[1:800],alpha=1,lambda=lambda.min)
lasso.pred=predict(lasso.mod,s=lambda.min,newx=x[801:1000])
```

We computed MSE_{lasso} which showed a great improvement compared to MSE_{ols}. In fact, we obtained MSE_{lasso}=0.2304754, whereas the MSE_{ols}= 0.4205839. Overall, we believe this proves the superiority of Lasso regression in a context where there is the presence of many predictors. Overall, from the 94 variables in the **X**, Lasso selected 33, as reported in Table 7.

Table 7

Variables selected by LASSO	Description
ASVABAR	Arithmetic reasoning
ASVABMK	Word knowledge
CATSE	Self-employment
COHABIT	Cohabiting
COLLBARG	Pay set by collective bargaining
EDUCBA	Bachelor's degree
EDUCMAST	Master's degree
EDUCPHD	Doctorate
EDUCPROF	Professional degree
ETHHISP	Ehtnicity, Hispanic
EXP	Total out-of-school work experience
FAITHJ	Faith, none
FEMALE	Sex of respondent (0 if male, 1 if female)
HEIGHT	Height, in inches
HHINC97	Gross household income,$, in year prior 1997
HOURS	Usual number of hours worked per week
JOBS	Number of jobs
MALE	Sex of respondned (0 if male, 1 if female)
MARRIED	Married,spouse present
MSA11NK	MSA, not known
MSA97NCC	MSA, not central city
MSA97NO	Not in metropolitan statistical area
OTHSING	Other single
REG97NC	Census Region north central
REG97NE	Census Region north east
REGNC	Census Region north central
REGS	Census Region south
S	Years of schooling (highest grade completed)
SF	Years of schooling of biological father
SFR	Years of schooling of residential father
SIBLINGS	Number of siblings
TENURE	Tenure (years) with current employer
WEIGHT11	Weight, in pounds

6 EAWE dataset: exploring Lasso extensions

In the following section, we will provide three illustrative examples of extensions to Lasso using the EAWE dataset. The aim is to show how these estimators work in practice and how different the results are from traditional Lasso. For the applications, we will use the EAWE dataset with the same subsets selected from EAWE01 and EAWE02, corresponding to a total of 1000 observations. We decided, when possible, to fit the models on the entire set of EAWE variables, as we did for Lasso.

6.1 Adaptive Lasso

To estimate the two-stage approach in Adaptive Lasso, we used the package `glmnet`, specifying the option `penalty.factor`.

First, we performed the same data splitting as we did for Lasso, then we estimated OLS to obtain the estimates "consistent and unbiased" to specify the adaptive lasso weights. As previously explained, to compute the weight we take the inverse of the absolute values of the OLS coefficients raised to the power γ. In our application, we considered four different γ values: 0.1, 0.5,1 and 2 to see how different γ values affected results from Adaptive Lasso. Below, we report the code to compute $\hat{\mathbf{w}}_\mathbf{j} = \frac{1}{|\hat{\beta}_j^{0.1}|}$. We used the specified weights both to conduct LOOCV and to fit the model. In the following lines of code we the command needed to estimate Adaptive Lasso. First we compute the adaptive lasso weights using previously estimated OLS coefficient using as γ value 0.1. Then we conduct LOOCV specifying the option `penalty.factor` which allow us to use the previously defined weights. Finally, we extracted the predicted values using the testing dataset.

```
weight1 <- 1/abs(matrix(best_ols_coef)
[, 1][2:(ncol(x1)+1)])^0.1
alasso1_cv <- cv.glmnet(x1,y,
type.measure = "mse",
nfold = 800, alpha = 1,
penalty.factor = weight1,
keep = TRUE,grouped=FALSE)
alasso.coef=predict(alasso1,
type="coefficients",s= alasso1_cv$lambda.min, newx = x2) [1:91,]
```

In Table 8 we report the different variables selected estimated for the different γ values. As the table shows the fewer variables were included for a value of γ=1 and for γ=0.1.

Moreover, we compute the out-of-sample MSE for each model estimated. As Table 9 shows, as γ increases the out-of-sample MSE increases, though the increases are quite small in magnitude.

6.2 Induced Smoothed Lasso

We performed `islasso` in R, using the package `islasso`, following Sottile et al. (2019). The package `islasso` returns point estimates, reliable standard errors, and corresponding p-values for the regression coefficients. We aimed to conduct Islasso on the entire X matrix of EAWE explanatory variables. However, unlike traditional Lasso, we had to remove the following variables—because multicollinearity—to make Islasso work: HHBMONLY, HHBFONLY, MSA97NK, RS97UNKN, MSA11NIC.

Table 8

γ=0.1	γ=0.5	γ=1	γ=2
ASVABAR	FEMALE	FEMALE	ASVABAR
ASVABMK	ASVABAR	ASVABMK	ASVABMK
ASVABPC	ASVABMK	ASVABPC	ASVABPC
CATPRI	ASVABPC	CATSE	ASVABWK
CATSE	ASVABWK	COLLBARG	BYEAR
COHABIT	BYEAR	EDUCAA	CATSE
COLLBARG	CATPRI	EDUCBA	COLLBARG
EDUCAA	CATSE	EDUCGED	EDUCAA
EDUCBA	COHABIT	EDUCMAST	EDUCBA
EDUCGED	COLLBARG	EDUCPHD	EDUCGED
EDUCMAST	EDUCAA	EDUCPROF	EDUCHSD
EDUCPHD	EDUCBA	ETHHISP	EDUCMAST
EDUCPROF	EDUCGED	EXP	EDUCPHD
ETHHISP	EDUCMAST	FAITHJ	EDUCPROF
EXP	EDUCPHD	HHBFONLY	ETHHISP
FAITHJ	EDUCPROF	HHBMBF	EXP
FEMALE	ETHHISP	HHBMONLY	FAITHC
HEIGHT	EXP	MARRIED	FAITHJ
HHBMBF	FAITHJ	MSA11NCC	FAITHO
HOURS	HEIGHT	MSA11NO	FEMALE
JOBS	HHBFONLY	MSA97CC	HHBFONLY
MARRIED	HHBMBF	MSA97NO	HHBMBF
MSA11NCC	HHBMONLY	REG97NE	HHBMOF
MSA11NO	HHOMBF	REGNC	HHBMONLY
MSA97NCC	HOURS	REGNE	HHOMBF
MSA97NO	JOBS	REGW	MARRIED
REG97S	MARRIED	RS97RURL	MSA11CC
REGNC	MSA11NCC	S	MSA11NCC
REGW	MSA11NO	TENURE	MSA11NO
RS97RURL	MSA97CC	**p=29**	MSA97CC
S	MSA97NO		MSA97NO
SF	PRFSTYAN		PRFSTYAN
SFR	REG97NE		PRFSTYPE
SIBLINGS	REGNC		REG97NE
TENURE	REGNE		REGNC
WEIGHT11	REGW		REGNE
p=38	RS97RURL		REGW
	S		RS97RURL
	SIBLINGS		S
	SINGLE		TENURE
	TENURE		URBAN
	URBAN		**p=41**
	p=42		

Table 9

Out of sample MSE

γ=0.1	0.2254022
γ=0.5	0.2277718
γ=1	0.2281781
γ=2	0.2333396

Once we removed these variables, we selected λ by LOOCV, as previously explained. Then, as suggested by Sottile et al. (2019), to make the algorithm work faster, we multiplied the λ_{min} by the number of observations.

```
model=with(EAWE,cv.glmnet(x[1:800,],y, standardize=TRUE,type.measure='mse',
nfolds=800,alpha=1, grouped=FALSE))
lambda=model$lambda.min
lambdalasso=lambda*800
```

Having selected λ_{min} we proceeded by fitting islasso on the training dataset:

```
islasso.mod=islasso(y~.,data=data.frame(x[1:800,]), lambda=lambda,family=gaussian)
```

As previously explained,`islasso` doesn't shrink the coefficients precisely to zero, but it returns the p-value from the output of the fitted model. Stepping back to Lasso, we recall that 33 X variables were selected (Table 7). Table 10 reports the 19 variables that Islasso reported as statistically significant. Moreover, we computed out-of-sample MSE and compared it with OLS MSE. As a result, we obtained 0.2435632 out-of-sample MSE for islasso, which shows an improvement with respect to 0.4204438 OLS.

Table 10

	Estimate	Std. Error	Df	z value	Pr(> |z|)
(Intercept)	-73.701212	41.270703	1	-1.786	0.074132.
FEMALE	-0.107792	0.055788	0.999	-1.932	0.053338.
MALE	0.000073	0.000036	0.001	2.028	0.042566*
BYEAR	0.037494	0.020811	0.999	1.802	0.071608.
AGE	-0.000026	0.000014	0.001	-1.826	0.067861.
S	0.059571	0.01597	1	3.73	0.000191 ***
EDUCMAST	0.192818	0.104526	0.955	1.845	0.065084 .
EDUCHSD	-0.024029	0.000235	0.997	-102.087	2E-16 ***
MARRIED	0.078924	0.043512	0.999	1.814	0.0697 .
FAITHP	-0.006592	0.000196	0.997	-33.671	2E-16***
HEIGHT	0.015658	0.006964	1	2.248	0.024546 *
REG97NE	0.156453	0.088152	0.922	1.775	0.07593 .
REG97S	-0.000506	0.000131	0.856	-3.856	0.000115***
JOBS	-0.013206	0.006276	1	-2.104	0.035369 *
HOURS	0.004228	0.001736	1	2.436	0.014849 *
TENURE	0.021939	0.008426	1	2.604	0.009222 **
CATSE	0.237506	0.069172	1	3.434	0.000596 ***
COLLBARG	0.254832	0.057437	1	4.437	0.00000913***
MSA11CC	-0.006041	0.000153	0.996	-39.575	2E-16 ***
EXP	0.043212	0.010169	1	4.249	0.00002144***
Signif. cod: 0 '***	' 0.001 '**' 0.	01 '*	' 0.05 '.'	0.1 ' ' 1	

6.3 Relaxed Lasso

To estimate Relaxed Lasso, we used the package `glmnet` specifying the option `relax = TRUE`. As explained above, the Relaxed Lasso is equivalent to estimating two consecutive Lasso models. The first step is needed to perform variable selection as an ordinary Lasso, and the second is for the shrinkage of coefficients. Essentially, the second step is necessary to ensure that we obtain the best parameters estimates, based on a prediction criterion, for the best set of predictors. As the code below reports, other than adding the option `relax = TRUE`, the procedure is identical to the traditional Lasso estimation. The major difference is that from the `cv.glmnet` object we extract not only λ_{min} but also γ_{min}. As for Lasso, the parameter λ controls for the variable selection part, whereas the parameter γ controls the shrinkage of coefficients. If γ=1, then Lasso and relaxed Lasso estimators are identical for $\gamma < 1$, the shrinkage of coefficients is reduced compared to ordinary Lasso estimation. `cv.glmnet` uses by default 5 values of γ: (0, 0.25, 0.5, 0.75, 1). The λ we obtained is λ_{min}=0.03289, whereas γ_{min}=0.25.

```
cv=(cv.glmnet(x1,y[1:800], standardize=TRUE,type.measure='mse',nfolds=800,
alpha=1,  grouped=FALSE, relax=TRUE))
relax.coef=predict(cv,type="coefficients",
```

```
s= 0.01714892, newx = x2,gamma="gamma.min",
lambda="lambda.min",alpha=1)[1:97,]
relax.coef[relax.coef!=0]
```

In Table 11 we reported the variables selected by relaxed Lasso. Moreover, we computed the out-of-sample MSE, which results in MSE=0.235308.

Table 11

Variables Selected by Relaxed Lasso	
ASVABAR	ASVABMK
ASVABMK	CATSE
COHABIT	COLLBARG
EDUCBA	EDUCMAST
EDUCPHD	EDUCPROF
ETHHISP	EXP
FAITHJ	FEMALE
HEIGHT	HHINC97
HOURS	JOBS
MALE	MARRIED
MSA11NK	MASA97NCC
MSA97NO	OTHSING
REG97NC	REG97NE
REGNC	REGS
S	SF
SFR	SIBLINGS
TENURE	WEIGHT11

6.4 Traditional Lasso vs. Lasso Extensions

In Table 12 we summarized the variables that each method included, adding a column to identify which ones were always selected across different estimation methods. It's important to emphasize that the variables chosen by Lasso and Relaxed Lasso are identical, as Relaxed Lasso is simply updating the estimated coefficients by estimating Lasso twice: the difference would be in the shrinkage of the selected variables but not in their inclusion or exclusion. For the Islasso, we reported the variables that resulted statistically significant after the estimation. The variables that were included for every estimation technique were:

- CATSE, self-employment
- COLLBARG, pay set by collective bargaining
- EDUCMAST, master's degree
- FEMALE, sex of respondent
- S, years of schooling (highest grade completed)
- TENURE, tenure (years) with current employer

Moreover, we also reported the MSE across the different estimation techniques. As Table 13 shows, Lasso and Islasso have similar MSE, and the lowest MSE is obtained with the adaptive procedure and $\gamma = 0, 1$. Overall, each model specification show improvement with respect to OLS.

Table 12

EAWE	Lasso	Islasso	Adaptive Lasso $\gamma = 0.1$	Adaptive Lasso $\gamma = 0.5$	Adaptive Lasso $\gamma = 1$	Adaptive Lasso $\gamma = 2$	Relaxed Lasso	Variables always included
AGE		X						
AGEMBTH								
ASVABAR	X		X	X		X	X	
ASVABC								
ASVABC4								
ASVABCS								
ASVABMK	X		X	X	X	X	X	
ASVABMV								
ASVABNO								
ASVABPC			X	X	X	X		
ASVABWK				X		X		
BYEAR		X		X		X		
CATGOV								
CATMIS								
CATNPO								
CATPRI			X	X				
CATSE	X	X	X	X	X	X	X	X
COHABIT	X		X	X			X	
COLLBARG	X	X	X	X	X	X	X	X
EDUCAA			X	X	X	X		
EDUCBA	X		X	X	X	X	X	
EDUCDO								
EDUCGED		X	X	X	X	X		
EDUCHSD						X		
EDUCMAST	X	X	X	X	X	X	X	X
EDUCPHD	X		X	X	X	X	X	
EDUCPROF	X		X	X	X	X	X	
ETHBLACK								
ETHHISP	X		X	X	X	X	X	
ETHWHITE								
EXP	X	X	X	X	X	X	X	
FAITHC						X		
FAITHJ	X		X	X	X	X	X	
FAITHM								
FAITHN								
FAITHO						X		
FAITHP		X						
FEMALE	X	X	X	X	X	X	X	X
HEIGHT	X	X	X	X			X	
HHBFONLY				X	X	X		
HHBMBF			X	X	X	X		
HHBMOF						X		
HHBMONLY				X	X	X		
HHINC97	X						X	
HHOMBF						X		
HHOTHER								
HOURS	X	X	X	X			X	
JOBS	X	X	X	X			X	
MALE	X	X					X	
MARRIED	X	X	X	X	X	X	X	
MSA11CC		X				X		
MSA11NCC			X	X	X	X		
MSA11NIC								
MSA11NK	X						X	
MSA11NO			X	X	X	X		
MSA97CC				X	X	X		
MSA97NCC	X		X				X	
MSA97NK								
MSA97NO	X		X		X	X	X	
OTHSING	X						X	
POVRAT97								
PRFSTYAE								
PRFSTYAN				X		X		
PRFSTYPE						X		
PRFSTYUN								
PRMONF								
PRMONM								
PRMSTYAE								
PRMSTYAN								
PRMSTYPE								
PRMSTYUN								
REG97NC	X		X	X		X	X	
REG97NE	X	X		X	X	X	X	
REG97S	X	X	X				X	
REG97W				X				
REGNC	X		X	X	X		X	
REGNE				X	X			
REGS	X						X	
REGW			X	X	X	X		
RS97RURL			X	X	X	X		
RS97UNKN								
RS97URBN			X					
S	X	X	X	X	X	X	X	X
SF	X		X				X	
SFR	X		X				X	
SIBLINGS	X		X	X			X	
SINGLE				X				
SM								
SMR								
TENURE	X	X	X	X	X	X	X	X
URBAN				X		X		
VERBAL								
WEIGHT04								
WEIGHT11	X		X			X		

Table 13

	MSE
OLS	0.4204438
Lasso	0.2304754
Islasso	0.243563
Adaptive $\gamma = 0,1$	0.2254022
Adaptive $\gamma = 0,5$	0.2277718
Adaptive $\gamma = 1$	0.2281781
Adaptive $\gamma = 2$	0.2333396
Relaxed Lasso	0.235308

7 Mtcars & EAWE LOOCV vs k-folds cross validation

Earlier, we emphasized using the LOOCV method to pick the Lasso penalty value. LOOCV's advantage is that two researchers would not obtain different results due solely to random sampling to construct the folds. As we underlined, another common approach is $k - folds$ cross-validation, where observations are divided into k folds: the first fold is treated as a validation set and the remaining $k - 1$ as the training set. The issue of which cross-validation method is best is not settled in the literature. James et al. (2013) underline that even if *LOOCV* gives unbiased test error estimates, it could have higher variance than $k - folds$ CV. Indeed, when we perform *LOOCV*, we average the outputs of n fitted models, trained with almost overlapping training sets leading to highly positively correlated outputs, thereby higher variance. If there is a spurious correlation in a training set, it's hard to determine a spurious and real correlation when refitting the model because the training set never changes. To summarize, according to James et al. in terms of bias reduction, *LOOCV* is preferred to $k - folds$, but not necessarily in terms of variance.

As we do not aim to settle the dispute among cross-validation methods, in the next section, we will offer a perspective on each technique, comparing their results. As we will be using a CV to pick $\lambda's$ we believe that LOOCV is a better technique for replicating the results.

7.1 LOOCV vs. k-folds cross validation with Mtcars

To perform cross-validation in R we will use the Mtcars dataset and the the `glmnet` package, specifically `cv.glmnet`. First we set $y = mpg$, specifying the `center=TRUE, scale=FALSE` option. Then, we created the matrix of explanatory variables X. Our aim is to perform the cross-validation procedure 10 times, to see how results for λ_{min} differ across replications. As noted earlier, the randomness in selecting the training and evaluation datasets means that running the command multiple times is likely to lead to multiple values of λ_{min}. First, we will proceed with cross-validation for Ridge regression, thereby setting $\alpha = 0$:

```
y <- mtcars %>% select(mpg) %>% scale(center=TRUE, scale=FALSE) %>% as.matrix()
X <- mtcars %>% select(-mpg) %>% as.matrix()
lambdas=matrix(-99,1,nreps) #create space to store lambda_{min}
cvm=matrix(-99,nreps)
for (iii in 1:nreps) {
cvridge=(cv.glmnet(X, y, alpha = 1,
standardize = TRUE, nfolds = 10, grouped=FALSE))
lambda=cvridge$lambda.min
I0=which(cvridge$lambda==cvridge$lambda.min)
lambdas[iii]=cvridge$lambda.min
cvm[iii]=cvridge$cvm[I0]
}
```

As shown, the first thing to do is to set a random seed: $k-folds$ cross-validation contains randomness, and setting a random seed allows us to replicate the results later. We first specified 10-folds cross-validation: this approach randomly divides the sample into ten folds of approximately equal size. The first fold is treated as a validation set, and the method is fit using data from the remaining nine folds. Then, the MSE is computed on the held-out observation. The procedure is repeated t times and each time a different group of observations is treated as a validation set. In other words, for each replication, we create different t cross-validation folds. λ_{min} is obtained by picking the value that minimize the CV measure calculated across all t-folds. As a result, for different replications since the folds are random, we obtain different values of λ_{min}. We reported the ten λ_{min} values obtained with 10-folds cv in Table 13.

We observe that λ_{min} values are repeated across replications, thus across different cross-validation folds. This is because `glmnet` performs Ridge Regression for an automatically selected range of λ. We performed the same procedure but with `nfolds=32` to perform LOOCV. In this case, we obtained the same λ_{min} across replications, as shown in the second column of Table 14.

To compare how different λ_{min} expected values affect model estimation, we fit a Ridge model for each of the λ_{min} values selected by cross-validation. Then, we extracted the estimated coefficients, $\hat{y}$, and the predictions and we computed the % change among coefficients fit between λ's value, keeping as s reference value the $\lambda_{minLOOCV}$. We considered the difference between $\lambda_{minLOOCV}$ and the smallest and largest λ_{min} obtained by k-folds cross validation. We observe that there is a large variability among the estimated values. On average, comparing the results using $\lambda_{minLOOCV}$ to those using λ_{min}=2.2804328 there is a 20% difference and comparing results from $\lambda_{minLOOCV}$ with those using λ_{min}=3.014598, there is an 11% difference.

Table 14

10foldsCV	LOOCV
3.014598	2.746789
2.746789	2.746789
3.014598	2.746789
2.502772	2.746789
3.014598	2.746789
3.014598	2.746789
2.502772	2.746789
2.280432	2.746789
2.280432	2.746789
2.502772	2.746789

Table 15

10 folds CVM	LOOCV CVM
7.349375	7.336113
7.259769	7.336113
6.905356	7.336113
8.244306	7.336113
9.359265	7.336113
7.990501	7.336113
7.122686	7.336113
7.042466	7.336113
7.894833	7.336113

Moreover, as shown in Table 14, from the object `cv.glmnet` we extracted the *cvm*, the mean cross-validated error value. As `cv.glmnet` fits the model for 100 λ, the *cvm* is a vector of the same length of λ so we extracted only the *cvm* value corresponding to each λ_{min}.

We repeated the same procedure but for Lasso Regression, thereby setting $\alpha = 1$. In Table 16 and Table 17, we report the λ_{min} selected from *LOOCV* and $10 - folds$ cross validation. Again, we observe that the $\lambda_{minLOOCV}$ value is in the middle of the range of $\lambda's$ selected by $10 - foldsCV$.

Table 16

10 foldsCV	LOOCV
0.6647582	0.7295713
0.8007036	0.7295713
0.7295713	0.7295713
0.6057029	0.7295713
0.6647582	0.7295713
0.7295713	0.7295713
0.6647582	0.7295713
0.4174875	0.7295713
0.4174875	0.7295713
0.458192	0.7295713

Table 17

10folds CVM	LOOCV CVM
7.949812	7.994084
8.403699	7.994084
7.628242	7.994084
8.083009	7.994084
8.687241	7.994084
9.989269	7.994084
8.287426	7.994084
7.393465	7.994084
7.898742	7.994084
8.845132	7.994084

We observe we observe that for Lasso there is less variability among the predicted values than Ridge Regression. On average, comparing the results using $\lambda_{minLOOCV}$ to those using λ_{min}=0.4174875 there is a 16% ; difference and comparing results from $\lambda_{minLOOCV}$ with those using λ_{min}=0.8007036, there is an 10% difference.

To conclude, we decided to use *LOOCV* motivated by the fact that there is no randomness in the data splits and replicability of results. Moreover, the high variability in the predicted $\hat{y}$ led us to say that to choose a method where the randomness is reduced to a minimum.

7.2 LOOCV vs. k-folds cross validation for EAWE

As for Mtcars, we conduced the same *LOOCV* vs. $k - folds$ CV for the EAWE dataset. Following the previous section we decided to use two different datasets to fit Ridge and Lasso.

For Ridge cross-validation we will use EARNINGS as response variables and S, ASVABC, ASVABC4, ASVABAR, ASVABWK as independent variables. Whereas for Lasso we will use the entire **X** matrix of independent variables available.

To perform cross-validation in R we will use the `glmnet` package, specifically the `cv.glmnet`. Our aim is to perform cross-validation procedure 10 times, to see how different results for λ_{min} across replications. First, we will proceed with cross-validation for Ridge regression, thereby setting $\alpha = 0$. From Table 18 we observe that $\lambda_{minLOOCV}$ is in the middle range of the λ_{min} selected by 10 folds cross validation. Moreover, as we did for Mtcars, we also fit a Ridge model for each of the λ_{min} selected by cross-validation. Then, we extracted the estimated coefficients $\hat{y}$, and we computed the % change among coefficients fit between λ's value, keeping as a base reference the $\lambda_{minLOOCV}$. We observe that for the EAWE dataset there is less variability among the estimated coefficients. On average between $\lambda_{minLOOCV}$- λ_{min}=0.01755245 there is a less than 1% difference and the same goes for between $\lambda_{minLOOCV}$-λ_{min}=0.05360264.

Table 18

10foldsCV	**LOOCV**
0.01755245	0.0254656
0.01755245	0.0254656
0.05360264	0.0254656
0.02794846	0.0254656
0.01755245	0.0254656
0.01755245	0.0254656
0.0306734	0.0254656
0.0306734	0.0254656
0.0306734	0.0254656
0.0306734	0.0254656

Table 19

10folds CVM	**LOOCV CVM**
0.2870256	0.2875486
0.2874746	0.2875486
0.2893539	0.2875486
0.2871482	0.2875486
0.2873814	0.2875486
0.2874786	0.2875486
0.2882417	0.2875486
0.2883017	0.2875486
0.288728	0.2875486
0.2879898	0.2875486

Moreover, we conducted the same analysis for Lasso regression. Again, we observe that for the EAWE dataset there is less variability among the estimated coefficients compared to Mtcars: on average between $\lambda_{minLOOCV}$- λ_{min}=0.017148922 there is a less than 1% difference and the same goes for $\lambda_{minLOOCV}$-λ_{min}=0.01882091.

Table 20

10 folds	LOOCV
0.01882091	0.01714892
0.01562545	0.01714892
0.01562545	0.01714892
0.01714892	0.01714892
0.01714892	0.01714892
0.01714892	0.01714892
0.01562545	0.01714892
0.01562545	0.01714892
0.01714892	0.01714892
0.01714892	0.01714892

Table 21

CVM 10 folds	LOOCV
0.243252	0.2432404
0.2421694	0.2432404
0.2451041	0.2432404
0.2457068	0.2432404
0.2463512	0.2432404
0.2457754	0.2432404
0.2421929	0.2432404
0.2473571	0.2432404
0.2455701	0.2432404
0.2420777	0.2432404

8 UC Davis transfer students: an application

8.1 Data Description

This work aims to assess UC Davis transfer student performance, measured by time to degree. Specifically, we will use a dataset composed of 4,091 observations, over 11 years, of transfer students who enrolled initially in either Managerial Economics or Economics. The 126 variables which make up the dataset will enable us to understand the reasons why some transfers student do not graduate two years after transfer. Although the sample size is large, the large number of possible explanatory variables suggest that the use of shrinkage estimators might be beneficial.

Data from UC System (https://www.universityofcalifornia.edu/infocenter/transfers-major), indicate that transfer enrollment for the Economics major decreased over time, whereas the enrollment in the Managerial Economics increased over time. Moreover, we observe in Table 22 that the admit rate has been around 76% for Economics and 77% for Managerial Economics, with a negative peak in 2020. Since 2020, referring to the class entering in Fall 2020, the pandemic surely reduced both applications and acceptance. However, from the perspective of tracking these students only through Fall 2021, they should not enter in analysis of completion given that it is not possible to finish a bachelor's degree in less than 2 years after transferring.

To be considered as a transfer student, a student "must have completed 60 semesters or 90 quarter of UC-transferable units at one (or more) California community college, by the end of the spring term before fall enrollment". Moreover, a student should earn at least a 2.40 GPA in UC-transferable classes to meet UC requirements. Specifically, UC Davis requires a minimum 2.80 GPA to be considered for admission.

Usually, students transfer at the end of their Sophomore year; therefore, they enter UC Davis in the fall of their third year (Junior year). In this work, we considered only students who started

Table 22

Economics

Year	Applicants	Admit	Enrolls	Admit rate
2013	1720	1372	347	80%
2014	1817	1381	372	76%
2015	1942	1434	398	74%
2016	2111	1684	429	80%
2017	1850	1507	358	81%
2018	1998	1395	308	70%
2019	1837	1308	285	71%
2020	1238	964	187	78%

Managerial Economics

Year	Applicants	Admit	Enrolls	Admit rate
2013	451	366	153	81%
2014	422	338	145	80%
2015	471	354	160	75%
2016	484	410	180	85%
2017	676	549	193	81%
2018	730	585	213	80%
2019	701	566	231	81%
2020	1554	864	190	56%

and ended their academic careers at a California community college before transferring. A number of students transferred to UC Davis with units from institutions other than California Community Colleges (CCCs). Even if such students had taken some units from CCCs, they were excluded from this analysis (286 observations).

To be considered graduating on time, students should graduate in two years: if a student enrolled in Fall 2014, she must graduate by Spring 2016. Nearly all students entered in the fall quarter. We dropped a small number of observations corresponding to students who entered in a different quarter other than the fall.

We analyzed the student's performance for two majors, Managerial Economics and Economics (Table 23). Overall in our sample 2,639 students enrolled as Economics majors, whereas 1,481 transfer students entered as Managerial Economics majors. We observe that from 2011 to 2021 the gap between those entering into Economics and Managerial economics shrunk.

Table 23

ENTRYMAJOR

2011	205	66%	105	34%	310
2012	255	65%	135	35%	390
2013	261	69%	115	31%	376
2014	275	72%	109	28%	384
2015	298	72%	123	29%	421
2016	300	68%	141	32%	441
2017	284	67%	138	33%	422
2018	238	58%	169	42%	407
2019	220	56%	171	44%	391
2020	148	51%	140	49%	288
2021	135	52%	126	48%	261
Overall	238.09	63%	133.82	37%	371.91

As we aim to understand the characteristics that increase the probability of late graduation by transfer students, we will proceed with a descriptive analysis of the variables that seem likely to predict a student's time to degree.

First, we analyzed how many students stayed in their major until graduation: we defined the binary variable "STAYED" to see whether students switched majors or not. As Table 24 shows, 7.50% of Economics students changed majors and almost 13% of Managerial Economics students did so, thereby showing a higher attrition rate. It is important to note that if the student switched between these two majors, then they would still count as "STAYED=1" because the two majors are sufficiently similar that switching between the two could presumably be expected to have no effect on the time to degree. Other students who transferred first onto other major (let's say sociology) and then changed major to Managerial Economics or Economics are not included in the analysis, and they count as STAYED=0. Similarly, we presumed that having more than one major could increase the probability

Table 24

Entering Class	Stayed=0				Stayed=1			
	Economics	%	Man Econ	%	Economics	%	Man Econ	%
2011	12	5.85%	6	10.81%	193	94.15%	99	94.29%
2012	16	6.27%	12	11.51%	239	93.73%	123	91.11%
2013	28	10.73%	5	20.29%	233	89.27%	110	95.65%
2014	20	7.27%	11	16.95%	255	92.73%	98	89.91%
2015	28	9.40%	9	19.72%	270	90.60%	114	92.68%
2016	27	9.00%	10	17.09%	273	91.00%	131	92.91%
2017	17	9.40%	14	12.06%	267	94.01%	124	89.86%
2018	17	7.14%	17	10.06%	221	92.86%	152	89.94%
2019	11	5.00%	12	6.47%	209	95.00%	159	92.98%
2020	7	4.73%	10	5.11%	141	95.27%	130	92.86%
Overall	18.3	7.48%	10.6	12.86%	230.1	92.86%	124	92.12%

of graduating late. As Table 25 shows only 3.89% of these Economics student graduated with a double major, whereas just 2% of Managerial Economics student have done so. Moreover, as Table 26 reports, we analyzed whether students had a minor. Again we presume that having a minor might increase the probability of graduating late. Overall, 25.64% of Economics students had a minor and 26.33% of Managerial Economics did so. In both cases, the reason is straightforward: the additional requirements for a second major or a minor could complicate the scheduling of required courses of lead to greater total units before graduation.

We also analyzed how many students benefited from the transfer admission guarantee (TAG) program. The TAG program provides community college students with a guarantee of admission to UC when they are ready to transfer. Six UC campuses (Davis, Irvine, Merced, Riverside, Santa Barbara, and Santa Cruz) offer guaranteed admission to students from any of the California community colleges These students must meet campus-specific requirements to qualify for a TAG. To be considered for a TAG at UC Davis, a student must have at least a 3.20 GPA. We presume that having a TAG would

Table 25

Entering Class	Double Major=1				Double Major=0			
	Econ	%	Man Econ	%	Econ	%	Man Econ	%
2011	9	4.39%	2	1.90%	196	95.61%	103	98%
2012	8	3.14%	1	0.74%	247	96.86%	134	99%
2013	11	4.21%	1	0.87%	250	95.79%	114	99%
2014	15	5.45%	3	2.75%	260	94.55%	106	97%
2015	13	4.36%	1	0.81%	285	95.64%	122	99%
2016	7	2.33%	3	2.13%	293	97.67%	138	98%
2017	10	3.52%	2	1.45%	274	96.48%	136	99%
2018	7	2.93%	1	0.59%	232	97.07%	168	99%
2019	10	4.55%	5	2.92%	210	95.45%	166	97%
2020	6	4.05%	2	1.43%	142	95.95%	138	99%
Overall	9.6	3.89%	2.1	2%	238.9	96.11%	132.5	98%

Table 26

Entering Class	Minor=1				Minor=0			
	Econ	%	Man Econ	%	Econ	%	Man Econ	%
2011	63	30.73%	16	15.24%	142	69.27%	89	84.76%
2012	75	29.41%	16	11.85%	180	70.59%	119	88.15%
2013	64	24.52%	18	15.65%	197	75.48%	97	84.35%
2014	54	19.64%	23	21.10%	221	80.36%	86	78.90%
2015	68	22.82%	21	17.07%	230	77.18%	102	82.93%
2016	63	21.00%	30	21.28%	237	79.00%	111	78.72%
2017	92	32.39%	31	22.46%	192	67.61%	107	77.54%
2018	60	25.21%	45	26.63%	178	74.79%	124	73.37%
2019	55	25.00%	37	21.64%	165	75.00%	134	78.36%
Overall	66	25.64%	26.33	19.21%	194	74.36%	108	80.35%

decrease the probability of graduating late, as TAG students should have a high GPA and meet stricter requirements. But, it is also possible that the TAG simply reflects better planning or advising. As we observe from Table 27, 35.35% of Economics students and 48.22% of Managerial Economics students benefited from a TAG, almost 42% of our sample.

Next, we analyzed how many students enrolled in the Educational Opportunity Program (EOP). The EOP provides admission, academic and financial support services to historically underserved students throughout California. Therefore, we can reasonably presume that EOP students come from underprivileged backgrounds. We presume that this might increase the probability of late graduation: Jury et al. (2017) stated that the higher education system seems to favor students with more privileged socioeconomic background bringing more burden to the students from lower socioeconomic backgrounds. EOP students may need to work to help finance their education and they are more likely to be the first in their family to attend college. Overall, as reported in Table 28, 12.21% of Economics students and 17.63% of Managerial Economics enrolled in EOP.

Another variable we deemed interesting to comment is how many students had to pay a non-residential supplemental tuition (NRST). The NRST is paid by all students who are not residents of California, including both residents of other states and international students. We presume that

Table 27

Entering Class	TAG=1				TAG=0			
	Economics	%	Man Econ	%	Economics	%	Man Econ	%
2011	104	50.73%	71	67.62%	101	49.27%	34	32.38%
2012	70	27.45%	64	47.41%	185	72.55%	71	52.59%
2013	68	26.05%	49	42.61%	193	73.95%	66	57.39%
2014	108	39.27%	49	44.95%	167	60.73%	60	55.05%
2015	83	27.85%	51	41.46%	215	72.15%	72	58.54%
2016	80	26.67%	62	43.97%	220	73.33%	79	56.03%
2017	80	28.17%	54	39.13%	204	71.83%	84	60.87%
2018	84	35.29%	78	46.15%	154	64.71%	91	53.85%
2019	90	40.91%	74	43.27%	130	59.09%	97	56.73%
2020	71	47.97%	76	54.29%	77	52.03%	64	45.71%
2021	52	38.52%	75	59.52%	83	61.48%	51	40.48%
Overall	80.91	35.35%	63.91	48.22%	157.18	64.65%	69.91	51.78%

Table 28

Entering Class	EOP=1				EOP=0			
	Economics	%	Man Econ	%	Economics	%	Man Econ	%
2011	26	12.68%	17	16.19%	179	87.32%	88	83.81%
2012	29	11.37%	25	18.52%	226	88.63%	110	81.48%
2013	32	12.26%	25	21.74%	229	87.74%	90	78.26%
2014	32	11.64%	13	11.93%	243	88.36%	96	88.07%
2015	38	12.75%	25	20.33%	260	87.25%	98	79.67%
2016	25	8.33%	22	15.60%	275	91.67%	119	84.40%
2017	32	11.27%	20	14.49%	252	88.73%	118	85.51%
2018	28	11.76%	32	18.93%	210	88.24%	137	81.07%
2019	31	14.09%	33	19.30%	189	85.91%	138	80.70%
2020	23	15.54%	25	17.86%	125	84.46%	115	82.14%
2021	17	12.59%	24	19.05%	118	87.41%	102	80.95%
Overall	28.45	12.21%	23.73	17.63%	209.64	87.79%	110.09	82.37%

paying more each quarter will decrease the probability of graduating late. As Table 29 shows, 31% of Economics students and 17% of Managerial are classified as non-California residents.

Lastly, we considered the satisfaction of the Intersegmental General Education Transfer Curriculum (IGETC) as an explanatory variable. IGETC is a series of courses that California CC students can complete to satisfy most freshman/sophomore-level general education requirements before transferring to UC. 91.30% of Economics students and 91.25% of Managerial Economics students self-declared having completed IGETC. Table 30 shows that 75.20% of Economics students and 80.18% of Managerial Economics students provided proof of completion of IGETC. In recently admitted students, it is possible that they simply not yet provided documentation at the time the dataset was assembled. But for students from earlier years it seems reasonable to presume that they instead needed to take the additional courses to complete GE requirements.

Table 29

Entering Class	NRST=1				NRST=0			
	Economics	%	Man Econ	%	Economics	%	Man Econ	%
2011	50	24.39%	21	15.56%	155	75.61%	114	84.44%
2012	86	33.73%	14	12.17%	169	66.27%	101	87.83%
2013	76	29.12%	24	22.02%	185	70.88%	85	77.98%
2014	80	29.09%	22	17.89%	195	70.91%	101	82.11%
2015	106	35.57%	29	20.57%	192	64.43%	112	79.43%
2016	116	38.67%	32	23.19%	184	61.33%	106	76.81%
2017	84	29.58%	25	14.79%	200	70.42%	144	85.21%
2018	79	33.19%	31	18.13%	159	66.81%	140	81.87%
2019	68	30.91%	27	19.29%	152	69.09%	113	80.71%
2020	38	25.68%	8	6.35%	110	74.32%	118	93.65%
Overall	78.3	30.99%	23.3	16.99%	170.1	69.01%	113.4	83.01%

Table 30

Entering Class	Student Indicates IGETC Completed				IGETC Verified			
	Economics	%	Man Econ	%	Economics	%	Man Econ	%
2011	194	94.6	95	90.5	157	76.6	65	61.9
2012	236	92.5	118	87.4	204	80	91	67.4
2013	241	92.3	98	85.2	209	80.1	74	64.3
2014	262	95.3	102	93.6	221	80.4	70	64.2
2015	287	96.3	109	88.6	239	80.2	73	59.3
2016	285	95	126	89.4	246	82	82	58.2
2017	267	94	130	94.2	228	80.3	91	65.9
2018	222	93.3	158	93.5	177	74.4	115	68
2019	211	95.9	152	88.9	158	71.8	98	57.3
2020	141	99.3	134	95.7	96	64.9	83	59.3
2021	133	98.5	122	96.8	35	25.9	40	31.7
Overall	225.36	91.30%	122.18	91.25%	179.09	75.20	80.18	59.77%

We will now analyze a set of variables that reflect students' math skills and abilities: we believe that a lack of previous math preparation would lead students to graduate later. We considered this set of binary variables:

- Apcalc: whether a student had taken an Advancement Placement course (AP) in calculus;
- Precalc: whether students took courses that could be considered pre-calculus;
- Furthcalc: whether students took calculus courses beyond one year of calculus;
- tookCALCTEST: whether students took the UCD math screening test needed to evaluate a student's math skills. Not every student has to take the test, only students who haven't taken Calculus yet. A test score lower than 30 disqualifies the student from MAT 16A, below 25 from MAT 12, pre-calculus. We presume that taking the test shows a lack of previous math preparation and therefore a higher probability of late graduation;
- met16A: this binary variable takes the value of 1 if the grade of the equivalent MAT16A was higher than 0. In other words, if the grade is higher than 0, we presume students met the

Calculus requirements for the two majors;

- tookSTA13: takes the value of 1 if students took an exam equivalent to Elementary Statistics (STA13);
- TookMAT16A: takes the value of 1 if students took a course equivalent to Math 16A (Short Calculus).

As Table 31 and Table 32 show, the math skills are similar between Economics and Managerial Economics. Interestingly, we observe that there is a gap between the students who took a course equivalent to MAT16A for both majors and those who met the MAT16A GPA requirement. This indicates that we should be careful when evaluating results: taking a course doesn't necessarily imply having proficient math skills.

Table 31

% of students by entry year	Economics						
	Apcalc	**Precalc**	**Furthcalc**	**Calctest**	**met16A**	**tookSTA13**	**TookMAT16a**
2011	3.41%	69.27%	14.15%	13.17%	51.71%	91.71%	88.78%
2012	7.06%	67.45%	10.20%	15.29%	57.65%	89.41%	88.24%
2013	5.36%	66.28%	11.49%	16.48%	60.92%	93.49%	91.57%
2014	4.36%	70.91%	13.82%	8.73%	65.82%	94.18%	93.82%
2015	5.37%	73.83%	10.07%	9.73%	62.08%	94.30%	93.29%
2016	4.36%	72.00%	9.33%	6.33%	63.67%	93.00%	95.00%
2017	4.58%	62.68%	13.38%	7.04%	65.85%	93.66%	93.66%
2018	4.20%	69.75%	12.61%	9.24%	67.65%	94.12%	93.70%
2019	6.82%	69.09%	13.64%	9.09%	62.73%	93.18%	94.09%
2020	6.08%	62.84%	15.54%	1.35%	72.97%	96.62%	98.65%
2021	8.15%	57.78%	20.00%	0.00%	83.70%	85.93%	98.52%
Overall	5.43%	67.44%	13.11%	8.77%	64.98%	92.69%	93.57%

Table 32

% of students by entry year	Managerial Economics						
	Apcalc	**Precalc**	**Furthcalc**	**Calctest**	**met16a**	**tookSTA13**	**TookMAT16A**
2011	3.81%	66.67%	7.62%	19.05%	55.24%	96.19%	85.71%
2012	8.89%	62.96%	6.67%	14.07%	62.96%	97.78%	88.89%
2013	4.35%	66.96%	10.43%	13.91%	67.83%	98.26%	89.57%
2014	6.42%	54.13%	3.67%	15.60%	66.06%	98.17%	87.16%
2015	4.07%	62.60%	2.44%	14.63%	52.85%	93.50%	87.80%
2016	9.22%	65.25%	7.09%	16.31%	60.99%	95.74%	87.94%
2017	3.62%	69.57%	6.52%	13.77%	51.45%	95.65%	87.68%
2018	4.14%	64.50%	5.92%	10.06%	57.99%	96.45%	89.94%
2019	8.77%	62.57%	5.85%	9.36%	57.31%	92.40%	92.98%
2020	10.00%	60.00%	7.14%	10.71%	74.29%	98.57%	90.00%
2021	8.73%	62.70%	10.32%	0.79%	82.54%	98.41%	98.41%
Overall	6.66%	63.52%	6.65%	12.22%	62.43%	96.33%	89.80%

Moreover, we checked how many students had already taken equivalent core prerequisite courses. We presume that the more classes transfer students had taken, the lower the probability of graduating late. We specified a set of binary variables (took course=1, 0 otherwise) for the most important classes:

- MAT16A-Short Calculus, MAT16B-Integral Calculus, MAT16C-Short Calculus;
- ECN1A-Principles of Microeconomics, ECN1B-Principles of Macroeconomics;
- MGT11A-Elementary Accounting, MGT11B-Managerial Accounting;
- ARE18: Business Law
- PLS21: Computers in Technology;
- ECS15: Introduction to Computers;
- ECS32A: Introduction to Programming:
- CMN1-Introduction to public speaking, CMN3 Interpersonal Communication Competence.

As table 33 shows, results differ much among classes. Most of the students took core Economics classes such as ECN1A and ECN1B, but few students took preparatory courses for Programming (ECS32A) and Computers (ECS15) in general. Across the three programming courses, only one of which is required, fewer than 50% have taken one of the three courses. Managerial Economics transfers are more likely to have completed courses in accounting (MGT11A,MGT11B) and business law(ARE18) which are required for Managerial Economics but not for Economics.

Table 33

Prerequisites	**Econ**	**Man Econ**
	Mean	Mean
ECN1A	98.77%	97.56%
ECN1B	99.18%	97.99%
CMN3	7.38%	10.53%
CMN1	41.77%	60.46%
PLS21	0.75%	1.38%
ECS15	13.97%	29.82%
ECS32A	19.36%	36.03%
MGT11A	61.34%	87.16%
MGT11B	33.54%	66.67%
ARE18	26.13%	55.64%

Table 34 compares students grouped by their TAG status. Again, we do not observe significant differences between the two majors regarding which courses they had taken. Finally, in Table 35, we reported GPA across Major for all students, TAG and EOP students. In general, TAG has a higher GPA than non-TAG students, and surprisingly, EOP students show a slightly higher GPA.

Table 34

Percent of Students Taking the Course	TAG	Non-Tag
ECN1A	99.00%	96.10%
ECN1B	99.20%	97.00%
CMN1	62.70%	58.60%
CMN3	12.60%	8.80%
PLS21	1.70%	1.10%
ECS15	36.70%	24.10%
ECS32A	40.50%	32.30%
MGT11A	90.10%	84.70%
MGT11B	71.80%	62.40%
ARE18	60.10%	51.90%
STA13	98.30%	93.30%
MAT16A	95.30%	84.60%
MAT16B	0.10%	0.30%
MAT16C	69.60%	51.70%

Table 35

	GPA	
	Economics	**Man Econ**
GPA	3.394901	3.439787
TAG	3.479046	3.540994
EOP	3.419049	3.465208

We will now descriptively analyze the relationship between the variables described so far and the students who graduate late. We defined two binary variables to describe late graduation: GLATE, a stricter measure not including "summer term" after senior year as graduating on time, and GLATE2, which includes "summer term" as graduating on time. From Table 36, we observe that if we consider GLATE 47.5% of Economics students and almost 49.9% of Managerial Economics students graduated late. Instead, if we consider GLATE2, 26.7% of Economics students and 27.4% of Managerial Economics students graduated late.

Table 36

Entering Class	**GLATE**			**GLATE2**		
	Economics	**Man Econ**	**Overall**	**Economics**	**Man Econ**	**Overall**
2011	0.4190	0.5073	0.4632	0.2381	0.2976	0.2678
2012	0.4963	0.5176	0.5070	0.2444	0.2902	0.2673
2013	0.5652	0.5441	0.5546	0.3739	0.2874	0.3306
2014	0.4771	0.5491	0.5131	0.2661	0.3018	0.2839
2015	0.4715	0.5134	0.4925	0.3008	0.2752	0.2880
2016	0.4539	0.4333	0.4436	0.2695	0.2600	0.2648
2017	0.4928	0.5176	0.5052	0.2536	0.2711	0.2624
2018	0.4497	0.4328	0.4412	0.2308	0.2395	0.2351
2019	0.4503	0.4318	0.4411	0.2339	0.2455	0.2397
Overall	0.4751	0.4941	0.4846	0.2679	0.2742	0.2711

Our results don't different much from those that the UC Information Center reports on the general population of transfer students https://www.universityofcalifornia.edu/infocenter/ug-out

comes. Assuming that a student transfers at the end of the sophomore year, they should graduate within two years. The results show that on average, from 2010-2018, 59% graduated on time, which is higher than what we obtained just for Economics and Managerial Economics.

In addition, these statistics show that late graduation is more common for transfer students than a regular freshmen. In fact, averaging results from 2010 to 2016, 69% graduated on time.More specifically,we know that the general graduation rate (2012-2015) for freshman enrolled in Managerial economics is 70.50%, whereas for transfer students it is 64.10%

In Table 37, we report how the previously discussed variables influence late graduation.

Table 37

	Economics	**Man Econ**
	GLATE	
TAG=1	44.98%	41.30%
EOP=1	52.01%	52.10%
Apcalc=0	49.66%	49.91%
Precalc=0	46.19%	45.84%
Furthcalc=0	50.30%	50.51%
Tookcalctest=1	76.13%	78.18%
met16A=0	56.74%	57.43%
TookSTA13=0	63.58%	45.36%
TookMATH16=0	71.28%	76.87%
MAJDUM=1	81.11%	84.21%
MINDUM=1	36.53%	38.40%
NRST=1	40.27%	39.25%
IGETCSELF=0	49.48%	46.97%
IGETCVER=0	44.59%	40.58%

8.2 Descriptive analysis

To understand why transfers student tends to graduate later than freshmen, we used the set of predictors reported in Table 38. As noted, there are both demographic and more academic-related variables. Summing up what we discussed in the descriptive data section of this work, the following are the main hypothesis we drew:

1. being a TAG student would decrease the probability of late graduation;
2. being an EOP student would increase the probability of late graduation;
3. a lack of solid math preparation would increase the probability of late graduation;
4. having two majors/a would increase the probability of late graduation;
5. paying a non-residential student fee would decrease the probability of late graduation;
6. not having completed equivalent core courses at community college increases the probability of late graduation;
7. switching majors increases the probability of late graduation.

Before implementing the shrinking estimators discussed in the previous chapters, we started with a more traditional stepwise regression approach. We carried out the stepwise selection, and we estimated explanatory Probit models in Stata. The stepwise selection is a method for selecting subsets of predictors. There are two types of selection:

- forward stepwise selection;
- backward stepwise selection.

To perform forward stepwise selection, we start from the null model ($p = 0$). Then, we consider all the possible single-variable models, choosing the model with the highest statistically significant variable, in other words selecting the model that does the best by itself. Next, we consider all possible 2-variable models that consider the regressor chosen in the first step and choose the one with the highest statistically significant variable. Finally, we continue the process until we consider all p-variable models that include regressors in step (p-1) until there are no statistically significant variables.

Instead, to perform backward stepwise selection, we start from the full model. First, we consider all possible p-1 variables models, excluding one variable at a time, choosing the model with the least statistically significant variables. Then we consider all the possible p-2 variables model, excluding only one variable from those chosen in the first step. We continue the process until each variable remaining is statistically significant.

Table 38

Predictors		Mean	Sd	Min	Max
PERMRES	1=California permanent resident, 0=otherwise	0.1144	0.3183	0	1
TAG	1=Student has a tag, 0=otherwise	0.3894	0.4877	0	1
RETURN	1=returning student, 0=otherwise	0.0939	0.2917	0	1
ATHLETE	1=Athlete, 0=otherwise	0.0056	0.0748	0	1
OSSJA (Office of students support and Judicial Affairs)	1=Imposed delay, 0=otherwise	0.0032	0.0563	0	1
HSCA	1=High School California, 0=otherwise	0.7707	0.4239	-1	1
HSUS	1=High School USA, 0=otherwise	0.0372	0.1892	0	1
HSENG	1=High School English Speaking country, 0=otherwise	0.0044	0.0662	0	1
HSCHINA	1=High School China, 0=otherwise	0.1474	0.3545	0	1
HSHK	1=High School Hong Kong, 0=otherwise	0.0308	0.1728	0	1
HSKOR	1=High School Korea, 0=otherwise	0.0095	0.0972	0	1
HSOTH	1=High School Other countries, 0=otherwise	0.0592	0.2359	0	1
MAJDUM	1=Double Major, 0=otherwise	0.0286	0.1667	0	1
MINDUM	1=More than one minor, 0=otherwise	0.2034	0.4026	0	1
IGETCVER	1=IGETC verified, 0=otherwise	0.6971	0.4596	0	1
EOP	1=student has EOP, 0=otherwise	0.1403	0.3473	0	1
FEMALE	1=student is female, 0=otherwise	-0.2664	2.3203	-9	1
NONCA	1=student is domestic but non Ca, 0=otherwise	0.0320	0.1761	0	1
NRST	1=if a student pay non residential student fee, 0=othewrise	0.2601	0.4387	0	1
FVISA	1=if the student has an F visa, 0=otherwise	0.2359	0.4246	0	1
year	Year of enrollment	2015.8630	2.9650	2011	2021
met16A	1=if student satisfied GPA requirement for 16A, 0=otherwise	0.6343	0.4817	0	1
TUNITSUSED	Transferrable Units-Units Lost	98.5831	7.8109	42	105.1
UNITSUSED	TUNITSUSED+Ap Credits	101.7422	8.4929	42	159
GPA	Student's GPA	3.4108	0.2677	2.68	4
STAYED	1=student didn't change major, 0=otherwise	0.9294	0.2563	0	1
TOOKECN1A	1=student took equivalent ECN1A, 0=otherwise	0.9841	0.1251	0	1
TOOKECN1B	1=student took equivalent ECN1B, 0=otherwise	0.9883	0.1077	0	1
TOOKCMN3	1=student took equivalent CMN3, 0=otherwise	0.0836	0.2768	0	1
TOOKCMN1	1=student took equivalent CMN1, 0=otherwise	0.4903	0.5000	0	1
TOOKPLS21	1=student took equivalent PLS21, 0=otherwise	0.0105	0.1020	0	1
TOOKECS15	1=student took equivalent ECS15, 0=otherwise	0.2063	0.4047	0	1
TOOKECS32A	1=student took equivalent ECS32A, 0=otherwise	0.2569	0.4370	0	1
TOOKMGT11A	1=student took equivalent MGT11A, 0=otherwise	0.7057	0.4558	0	1
TOOKMGT11B	1=student took equivalent MGT11B, 0=otherwise	0.4432	0.4968	0	1
TOOKARE18	1=student took equivalent ARE18, 0=otherwise	0.3627	0.4809	0	1
TOOKSTA13	1=student took equivalent STA13, 0=otherwise	0.9413	0.2350	0	1
TOOKMAT16A	1=student took equivalent MAT16A, 0=otherwise	0.9201	0.2712	0	1
TOOKMAT16B	1=student took equivalent MAT16B, 0=otherwise	0.0005	0.0221	0	1
TOOKMAT16C	1=student took equivalent MAT16C, 0=otherwise	0.6355	0.4813	0	1
Apmicro	1=student has AP micro, 0=otherwise	0.0193	0.1376	0	1
Apmacro	1=student has AP macro, 0=otherwise	0.0178	0.1324	0	1
Apstat	1=student has AP statistics, 0=otherwise	0.0325	0.1774	0	1
Apcalc	1=student has AP calculus, 0=otherwise	0.0577	0.2332	0	1
ENTRYMAJOR	76=Economics, 77=Managerial Economics	76.3598	0.4800	76	77

We shall remember that as opposed to best subset selection, despite being more computationally feasible, stepwise doesn't consider all possible models, but only some combinations. Thereby it is not guaranteed it will give the best result. Other than that, there are some significant shortcomings embedded in stepwise regression that have been widely emphasized in the literature.

Standard statistical testing is usually used to test a hypothesis on a pre-specified model. It is not valid for stepwise regression as variables are selected in a series of steps (Smith, 2018). At each step, the variables with the smallest p-value are included in the model. This implies that the p-values of the variables left in the model are typically much smaller than they would be if we'd fitted a single model.

We carried out forward stepwise selection in Stata, specifying different level of significance.

```
stepwise , pe(.01): probit GLATE $predictors
```

In table 39, we report the variables included by the stepwise procedure following the different levels of significance. Moreover, the Stata command allowed us to fit a Probit model for each subset of

variables selected by the stepwise procedure. The choice of starting with the stepwise procedure is motivated by our will to show why shrinking estimators are a better option for variable selection.

Table 39

stepwise, pr(.01)	**stepwise,pe(.05)**	**stepwise,pe(.10)**	**stepwise,pe(.15)**	**stepwise,pe(.20)**	**stepwise, pe(.25)**
TOOKMAT16A	GPA	FEMALE	FEMALE	HSCA	FEMALE
STAYED	IGETCVER	GPA	GPA	GPA	GPA
MAJDUM	MAJDUM	IGETCVER	IGETCVER	IGETCVER	IGETCVER
TOOKMGT11B	MINDUM	MAJDUM	MAJDUM	MAJDUM	MAJDUM
GPA	NRTS	MINDUM	MINDUM	MINDUM	MINDUM
UNITUSED	STAYED	NRST	NRST	UNITUSED	NRST
NRST	TOOKMAT16A	STAYED	STAYED	STAYED	STAYED
TOOKMAT16C	TOOMKAT16C	TOOKMAT16A	TOOKECN1B	NRST	TOOKECMN3
MINDUM	TOOKMGT11B	TOOKMAT16C	TOOKMAT16A	HSUS	TOOKECN1B
IGETCVER	TOOKSTA13	TOOKMGT11B	TOOKMAT16C	HSHK	TOOKMAT16A
	UNITSUSED	TOOKSTA13	TOOKMGT11B	TOOKMAT16C	TOOKMAT16C
		UNITUSED	TOOKSTA13	PERMRES	TOOKMGT11A
			UNITUSED	TOOKMGT11B	TOOKMGT11B
				TOOKMAT16A	TOOKSTA13
				HSCHINA	UNITUSED
				HSENG	
				Apcalc	
				TOOKECN18	
				TOOKSTA13	
				FEMALE	

8.3 Does Lasso improve our model predictive power?

As we discussed in the data description section, we have the availability of 30 variables, some more related to students' characteristics. In contrast, others are more strictly related to students' academic performances. We first decided to subset the variables into academic and personal and carry out Lasso on this subset. We decided to conduct a preliminary analysis on these two subsets of variables as we were interested in analyzing whether academic or personal characteristics are more important to predict transfer's late graduation. We considered the sets of variables reported in Table 40.

Table 40

Academic Variables	**Personal Variables**
MAJDUM	PERMRES
MINDUM	TAG
IGETCVER	RETURN
MET16A	ATHLETE
TUNITUSED	FEMALE
UNITUSED	HSCA
GPA	HSUS
STAYED	HSENG
TOOKECN1A	HSCHINA
TOOKECN1B	HSHK
TOOCMN3	HSKOR
TOOKCMN1	HSOTH
TOOKPLS21	EOP
TOOKECS15	NONCA
TOOKECS32A	FVISA
TOOKMGT11A	year
TOOKMGT11B	NRST
TOOKARE18	MILITARY
TOOKSTA13	
TOOKAMT16A	
TOOKMATH16B	
TOOKAMAT16C	
Apmicro	
Apmacro	
Apstat	
Apcalc	
ENTRYMAJOR	
OSSJA	

We start with the analysis of academic variables. Before proceeding with the application of Lasso, we fit a Probit model as on the entire set of academic variables. We will use this model as a baseline to see whether the shrinking techniques will improve the model's predictive power. We report the results in Table 41. As shown, MAJDUM, MINDUM, UNITUSED, GPA, STAYED, TOOKECN1B, TOOKMGT11B, TOOKMAT16A, TOOKMAT16C are statistically significant. The results show that students who have two majors (MAJDUM) and took ECN1B (TOOKECN1B) are more likely to graduate late. The other significant variables have a negative direction, decreasing the probability of late graduation. As the conventional R^2 measures are problematic for models with a dichotomous dependent variable, to assess the validity of the Probit model we computed a confusion matrix, thereby measuring the proportion of correct predictions. We classified the observations as a "success", if $\hat{y}_i > 0.5$, $\hat{y}_i$=late graduation, and as a "failure" otherwise.

Table 41

	Dependent variable
	Y
MAJDUM	1.170*** (0.173)
MINDUM	−0.167*** (0.060)
IGETCVER	−0.403*** (0.060)
met16A	−0.025 (0.058)
TUNITSUSED	−0.006 (0.005)
UNITSUSED	−0.010** (0.005)
GPA	−1.002*** (0.104)
STAYED	−0.796*** (0.094)
TOOKECN1A	0.045 (0.221)
TOOKECN1B	0.504** (0.255)
TOOKCMN3	−0.067 (0.100)
TOOKCMN1	−0.007 (0.054)
TOOKPLS21	−0.397 (0.346)
TOOKECS15	−0.068 (0.065)
TOOKECS32A	−0.006 (0.063)
TOOKMGT11A	−0.088 (0.065)
TOOKMGT11B	−0.183*** (0.060)
TOOKARE18	0.052 (0.061)
TOOKSTA13	−0.134 (0.111)
TOOKMAT16A	−0.229** (0.106)
TOOKMAT16B	−4.294 (57.936)
TOOKMAT16C	−0.291*** (0.057)
APmicro	−0.099 (0.258)
APmacro	0.248 (0.264)
Apcalc	−0.159 (0.125)
APstat	−0.016 (0.161)
ENTRYMAJOR	0.016 (0.061)
OSSJA	0.750 (0.656)
Constant	4.907 (4.711)
Observations	2833
Log Likelihood	-1712.270
Akaike Inf. Crit.	3842.540
Note:	*p<0.1; **p<0.05; ***p<0.01 , SE in parentheses

Overall, as Table 42 shows, we obtained 11.5% of false-negative predictions (type 2 errors) and 20.7% of false-positive predictions (type 1 errors). The misclassification rate (percentage of total incorrect classifications made by the model) is 32.44%.

Table 42

	0	1
0	323	147
1	82	157

Next, we proceeded with estimation using Lasso. For this purpose, we split the sample into training and test data. As have 3542 observations and we proceeded with an 80-20 approach, we had 2833 observations in the validation sample and 710 in the test sample. First, we conducted LOOCV to select the λ_{min}. The λ_{min} selected was 0.00431814.

```
lasso.mod <- cv.glmnet(X1[1:2833,],Y[1:2833],
family = binomial(link =  "probit"),alpha=1,nfolds=2833)
```

As in previous analyses, we found that λ_{min} selected through LOOCV was in the middle range of the λ's selected by 10-folds CV, as shown in Table 43.

```
cv=lapply(1:10, function(i) {cv.glmnet(X1[1:2833,],Y[1:2833],
alpha=1, standardize=TRUE, type.measure='mse',
nfolds=10,grouped=FALSE) })
```

Table 43

10 folds CV	**LOOCV**
0.004728778	0.00431814
0.004728778	0.00431814
0.006251166	0.00431814
0.004728778	0.00431814
0.003577147	0.00431814
0.004308686	0.00431814
0.004728778	0.00431814
0.004728778	0.00431814
0.003925914	0.00431814
0.004728778	0.00431814

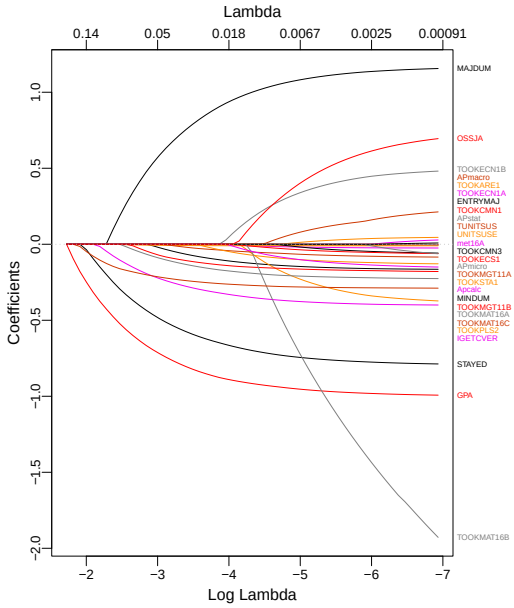
We proceeded with the Lasso estimation.

```
lasso.coef=predict(lasso.mod,type="coefficients",
s=lambdaloocv,newx=X1[2834:3542,])[1:29,]
lasso.coef[lasso.coef!=0]
```

In Figure 3. we report the coefficients path. In this plot, each colored line represents the value taken by each coefficient according to different values of λ. We observe that some coefficients go precisely to zero for specific values of λ. The larger the λ, the more coefficients are shrunk to zero. Still, even with large λ, some keep having a value different from zero, meaning that they

are valuable for predicting graduating late. Looking at the graph, we concluded that MAJDUM, OSSJA, TOOKECN1B, TOOKCMN1, STAYED, GPA, TOOKMAT16B are the most important variables in predicting graduating late.

Figure 3



In Table 44 we report the variables selected. We observe that out of 28 variables, Lasso procedure selected 22 variables.

Table 44

Variables selected by Lasso
MAJDUM
MINDUM
IGETCVER
met16A
TUNITUSED
UNITUSED
GPA
STAYED
TOOKECN1B
TOOKECMN3
TOOKPLS21
TOOKECNS15
TOOKMGT11A
TOOKMGT11B
TOOKARE18
TOOKSTA13
TOOKMAT16A
TOOKMAT16B
TOOKMAT16C
Apmacro
Apcalc
OSSJA

To sum up our results, in Table 45 we report the Probit and Lasso model AIC's,the smallest is obtained with Lasso.

Table 45

Model	**AIC**
Probit	3842,54
Lasso	3540,48

Moreover, we also computed the confusion matrix for the Lasso predicted values (Table 46) and its misclassification rate. We obtained 11.1% of false-negative predictions (type 2 errors) and 20.4% of false-positive predictions (type 1 errors). Overall, the misclassification rate (percentage of total incorrect classifications made by the model) is 31.59%, thereby showing improvement compared to the Probit misclassification rate 32.44%.

Table 46

	0	1
0	325	145
1	79	160

In a second step, we conducted the analysis for the personal variables. We conducted the same procedure as above. First, we fit a Probit model with the entire set of personal variables. Overall, as the confusion matrix reported in Table 48 shows, we obtained 19.3% of false-negative predictions (type 2 errors) and 24.4% of false-positive predictions (type 1 errors). The misclassification rate is 43.66%. We observe that the classification rate when we include only personal variables is more than 10% higher compared to the model including only academic variables.

Table 47

	Dependent variable
	Y
PERMRES	−0.265*** (0.090)
TAG	−0.248*** (0.051)
RETURN	0.003 (0.084)
FEMALE	−0.132** (0.052)
ATHLETE	0.049 (0.287)
HSUS	0.423* (0.243)
HSENG	0.319 (0.324)
HSCHINA	−0.284** (0.115)
HSHK	−0.187 (0.161)
HSKOR	1.026*** (0.278)
HSOTH	−0.049 (0.134)
EOP	−0.008 (0.074)
NONCA	0.109 (0.380)
FVISA	0.265 (0.345)
year	−0.014 (0.012)
NRST	−0.554 (0.338)
MILITARY	4.035 (62.957)
Constant	28.896 (23.819)
Observations	2,833
Log Likelihood	-1,894.853
Akaike Inf. Crit.	3,825.706
Note:	*p<0.1; **p<0.05; ***p<0.01

Table 48

	0	1
0	268	173
1	137	132

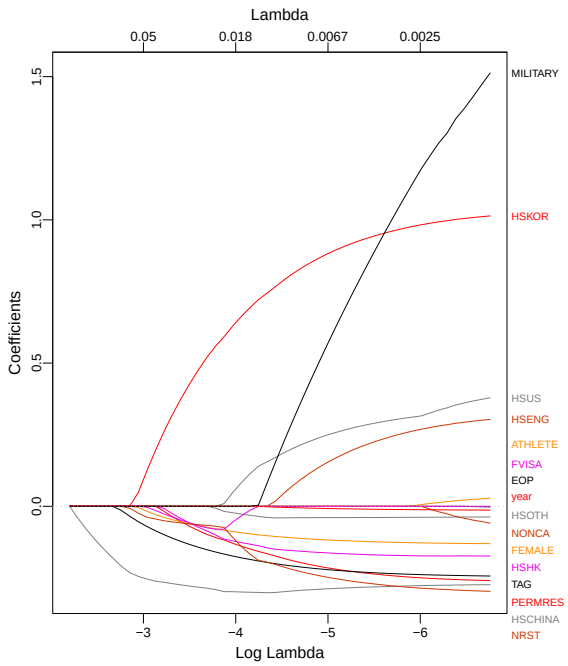
Next, we started conducing the procedure to fit Lasso. The λ_{LOOCV} identified was 0.003548358. As before, we also conducted 10 folds-CV to see whether λ_{LOOCV} was falling in the middle range. As Table 49 shows also in this case λ_{LOOCV} falls in the middle range of the λ's selected by 10-folds CV.

Table 49

10 foldsCV	**LOOCV**
0.003540590	0.00354059
0.003226054	0.00354059
0.003540590	0.00354059
0.004264653	0.00354059
0.004680451	0.00354059
0.003885793	0.00354059
0.003885793	0.00354059
0.002678326	0.00354059
0.002939460	0.00354059
0.004264653	0.00354059

In Figure 4, we report the path for the academic variables coefficients. As the figure shows MILITARY, HSKOR,HSUS,HSENG,FEMALE, HSHK, TAG, PERMRES,HSCHINA, NRST are the most important variables in predicting graduating late.

Figure 4



In Table 50 we report the variables selected by Lasso fit with λ_{LOOCV}.

Table 50

Variables selected by Lasso
PERMRES
TAG
FEMALE
HSUS
HSENG
HSCHINA
HSHK
HSKOR
HSOTH
year
NRST
MILITARY

To sum up our results in Table 51 we report the different models AIC. We observe that AIC's are similar across model and we do not observe a significant improvement across models.

Table 51

Model	**AIC**
Full Probit	3825,706
Lasso	3836,441

Moreover, we also computed the confusion matrix for the Lasso predicted values (Table 52) and its misclassification rate. We obtained 19.6% of false-negative predictions (type 2 errors) and 24.4% of false-positive predictions (type 1 errors). Overall, the misclassification rate (percentage of total incorrect classifications made by the model) is 44.01%, which is higher than the Probit misclassification rate.

Table 52

	0	1
0	315	136
1	90	169

Comparing the confusion matrices and the misclassification rates, we conclude that the subset of academic variables predicts the probability of graduation late better than the subset of personal variables.

After conducting a separate analysis for these two subset of variables, we decided to perform Lasso on a merged dataset in which we combined personal and academic variables (i.e., variables reported in Table 40). First, we run a Probit model on the entire set of variables. The confusion matrix reported in Table 54 shows that we obtained 13% of false-negative predictions (type 2 errors) and 17.3% of false-positive predictions (type 1 errors). The misclassification rate is 31.83%.

Table 53

	Dependent variable:
	Y
MAJDUM	1.176*** (0.175)
MINDUM	−0.192*** (0.062)
IGETCVER	−0.366*** (0.061)
met16A	0.0001 (0.059)
TUNITSUSED	−0.003 (0.006)
UNITSUSED	−0.014*** (0.005)
GPA	−0.996*** (0.111)
STAYED	−0.772*** (0.097)
TOOKECN1A	0.028 (0.228)
TOOKECN1B	0.525** (0.258)
TOOKCMN3	−0.086 (0.101)
TOOKCMN1	−0.067 (0.056)
TOOKPLS21	−0.403 (0.345)
TOOKECS15	−0.047 (0.067)
TOOKECS32A	−0.016 (0.064)
TOOKMGT11A	−0.046 (0.067)
TOOKMGT11B	−0.228*** (0.062)
TOOKARE18	−0.005 (0.063)
TOOKSTA13	−0.069 (0.113)
TOOKMAT16A	−0.241** (0.107)
TOOKMAT16B	−4.650 (92.126)
TOOKMAT16C	−0.255*** (0.059)
APmicro	−0.124 (0.259)
APmacro	0.249 (0.264)
APstat	−0.044 (0.163)
Apcalc	−0.209* (0.126)
ENTRYMAJOR	−0.001 (0.063)
OSSJA	0.897 (0.667)
PERMRES	−0.270*** (0.097)
TAG	−0.009 (0.057)
RETURN	−0.049 (0.091)
ATHLETE	−0.315 (0.330)
FEMALE	−0.070 (0.055)
HSCA	−1.173*** (0.287)
HSUS	−0.682* (0.351)
HSENG	−0.809* (0.424)
HSCHINA	−1.291*** (0.272)
HSHK	−1.253*** (0.294)
HSOTH	0.215 (0.143)
EOP	0.075 (0.079)
NONCA	−0.121 (0.396)
FVISA	−0.061 (0.364)
year	0.005 (0.013)
NRST	−0.325 (0.357)
MILITARY	3.755 (64.784)
Constant	−2.330 (27.409)
Observations	2,833
Log Likelihood	-1,674.783
Akaike Inf. Crit.	3441.566
Note:	*p<0.1; **p<0.05; ***p<0.01

Table 54

	0	1
0	313	123
1	92	182

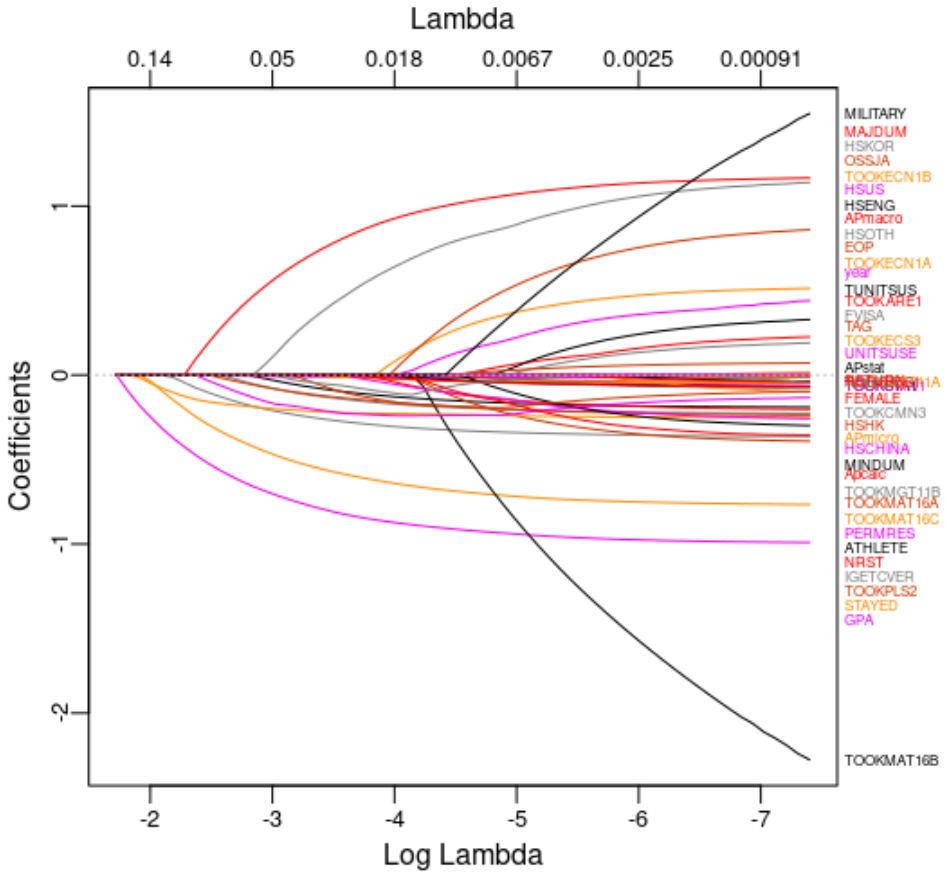
Then we started with the procedures needed to fit Lasso. We conducted CV to select λ_{min}. As Table 55 shows,λ_{loocv} was in the middle of the λ 's selected by 10-folds CV.

Table 55

10 folds CV	**LOOCV**
0.005695830	0.005201214
0.005695830	0.005201214
0.005695830	0.005201214
0.005695830	0.005201214
0.005189828	0.005201214
0.003925914	0.005201214
0.006251166	0.005201214
0.006251166	0.005201214
0.005189828	0.005201214
0.005695830	0.005201214

In Figure 4. we report the coefficients path. Looking at the graph, we concluded that MAJDUM, MILITARY, HSKOR, TOOKCMN1, STAYED, GPA, TOOKMAT16B, TOOKMGT11B,HSUS, TUNITUSED are the most important variables in predicting graduating late.

Figure 4



In Table 56 we report the variables selected by Lasso.

Table 56
Variables selected by Lasso

MAJDUM	PERMRES
MINDUM	ATHLETE
IGETCVER	FEMALE
TUNITSUSED	HSUS
UNITSUSED	HSENG
GPA	HSCHINA
STAYED	HSHK
TOOKECN1B	HSKOR
TOOKCMN3	HSOTH
TOOKCMN1	EOP
TOOKPLS21	FVISA
TOOKECS15	NRST
TOOKMGT11A	MILITARY
TOOKMGT11B	
TOOKSTA13	
TOOKMAT16A	
TOOKMAT16B	
TOOKMAT16C	
Apmacro	
Apcalc	
OSSJA	

To sum up our results in Table 57 we report the different models AIC. We observe that AIC's are similar across model and we do not observe a significant improvement across models.

Table 57

Model	**AIC**
Probit	3441.566
Lasso	3503.499

Finally, we computed the confusion matrix for the Lasso predicted values (Table 58) and its misclassification rate. We obtained 11.8% of false-negative predictions (type 2 errors) and 19.7% of false-positive predictions (type 1 errors). Overall, the misclassification rate (percentage of total incorrect classifications made by the model) is 31.97%.

Table 58

	0	1
0	321	143
1	84	162

In table 59 we reported the coefficient selected across the three different Lasso specification.

Table 59

X variables	Academic Lasso	Personal Lasso	Full Lasso
Apcalc	X		X
Apmacro	X		X
Apmicro			
Apstat	X		
ATHLETE			X
ENTRYMAJOR			
EOP			X
FEMALE		X	X
FVISA			X
GPA	X		X
HSCA			
HSCHINA		X	X
HSENG		X	X
HSHK		X	X
HSKOR		X	X
HSOTH		X	X
HSUS		X	
IGETCVER			X
MAJDUM	X		X
MET16A	X		
MILITARY		X	X
MINDUM	X		X
NONCA			
NRST		X	X
OSSJA	X		
PERMRES		X	X
RETURN			
STAYED	X		X
TAG		X	
TOOCMN3	X		X
TOOKAMAT16C	X		X
TOOKAMT16A	X		X
TOOKARE18	X		X
TOOKCMN1			X
TOOKECN1A			
TOOKECN1B	X		X
TOOKECS15	X		
TOOKECS32A			
TOOKMATH16B			
TOOKMGT11A	X		X
TOOKMGT11B	X		X
TOOKPLS21	X		X
TOOKSTA13	X		X
TUNITUSED			X
UNITUSED			X
year		X	

8.4 How to conduct inference on Lasso estimated coefficients: Islasso

Until now, we have stuck to the discussion on whether Lasso can improve predictive power. We will now dedicate the last section to see how to conduct inferences on the estimated coefficients. As explained, model selection and inference have long been considered conflicting. Fitting an unpenalized model or assessing coefficient significance after conducting Lasso it's a highly debated question: it is often considered cheating as we are peeking at the data twice. As a result, p-values and usual conference intervals are not valid. Therefore, rather than re-fitting an unpenalized model after Lasso we decided to adopt a different strategy. Researchers are debating on the best way to solve the inference-selection gap. The previous section presented different extensions to Lasso (adaptive, relaxed, and Islasso), allowing practitioners to conduct hypothesis testing on Lasso's estimated parameters. Another common approach is Belloni and Chernozhukov's OLS post-Lasso estimator (Belloni and Chernozhukov, 2013). If the sparsity assumption holds, the method allows fitting a naive post-Lasso with the variables selected by Lasso. In this work, we decided to apply Induced smoothed Lasso, as described in section 2.3.2, to conduct inference on Lasso estimated coefficients. Islasso enables us to conduct inference directly on the coefficients estimated by Lasso

```
islasso.mod=islasso(Y[1:2833]~.,X11[1:2833,] ,alpha=1,lambda=bestlambda,
family=binomial)
```

To perform islasso, we followed the same steps as Lasso. Therefore, we started with the academic variables analysis. Comparing Table 44 and Table 60 we observe some similarities between the coefficients selected by Lasso and those reported as significant by Islasso. Overall, comparing the two methods, we observe that Islasso reports fewer coefficients as significant than the coefficients selected by traditional Lasso.

We computed the confusion matrix for Islasso and the misclassification rate. We obtain almost 11.7% of false-negative predictions (type 2 errors) and 20.2% of false-positive predictions (type 1 errors). The misclassification rate is 31.31%.

Table 60

	Estimate	St. Error	Df	z value	Pr(> |z|)
Intercept	9.204.644	3.992.281	1.000	2.306	0.02113 *
MAJDUM	1.038.962	0.239861	0.963	4.332	1.48e-05 ***
MINDUM	-0.213988	0.092921	0.962	-2.303	0.02128 *
IGETCVER	-0.523862	0.094139	0.995	-5.565	2.63e-08 ***
met16A	-0.075473	0.068629	0.752	-1.100	0.27145
TUNITSUSED	-0.009442	0.007114	0.985	-1.327	0.18444
UNITSUSED	-0.018709	0.006676	0.999	-2.802	0.00507 **
GPA	-1.293.819	0.160780	0.984	-8.047	8.47e-16 ***
STAYED	-1.021.564	0.148813	0.986	-6.865	6.66e-12 ***
TOOKECN1A	-0.000001	0.050522	0.000	0.000	0.99998
TOOKECN1B	0.000059	0.067934	0.000	0.001	0.99931
TOOKCMN3	-0.000041	0.058788	0.001	-0.001	0.99945
TOOKCMN1	0.000037	0.000299	0.009	0.125	0.90021
TOOKPLS21	-0.000048	0.121895	0.000	0.000	0.99969
TOOKECS15	-0.025411	0.055355	0.553	-0.459	0.64619
TOOKECS32A	0.000016	0.052597	0.001	0.000	0.99976
TOOKMGT11A	-0.092253	0.076759	0.754	-1.202	0.22942
TOOKMGT11B	-0.226580	0.088514	0.978	-2.560	0.01047 *
TOOKARE18	0.018272	0.053430	0.566	0.342	0.73236
TOOKSTA13	-0.026148	0.053158	0.286	-0.492	0.62280
TOOKMAT16A	-0.188966	0.128517	0.731	-1.470	0.14146
TOOKMAT16B	-0.000034	0.372973	0.000	0.000	0.99993
TOOKMAT16C	-0.447193	0.087674	0.995	-5.101	3.38e-07 ***
APmicro	-0.000002	0.077489	0.000	0.000	0.99998
APmacro	0.000023	0.047952	0.000	0.000	0.99961
APstat	-0.000002	0.056875	0.000	0.000	0.99997
Apcalc	-0.008023	0.043033	0.185	-0.186	0.85211
ENTRYMAJOR	-0.000020	0.051586	0.001	0.000	0.99969
OSSJA	0.000039	0.139069	0.000	0.000	0.99978
Significance Codes:	0 ***	0.001 **	0.01 *	0.05 .	

Table 61

	0	1
0	326	144
1	78	161

Then, we estimated Islasso with the subsample of personal variables. We reported the results in Table 62. Comparing Table 52 and Table 62 we observe that Lasso and Islasso partially agree with the coefficients selected and reported as significant. Again, Islasso reports fewer coefficients as significant than those selected by Lasso, but overall the two methods lead to similar results. As for the other model specifications in Table 63 we computed the confusion matrix for Islasso and the misclassification rate we obtain almost 19.1% of false-negative predictions (type 2 errors) and 24.3% of false-positive predictions (type 1 errors). The misclassification rate is 43.58%.

Table 62

	Estimate	St. Error	Df	z value	$Pr(>\|z\|)$
Intercept	45.642.176	43.298.163	1.000	1.054	0.291820
PERMRES	-0.422325	0.164797	1.000	-2.563	0.010386*
TAG	-0.399764	0.092011	1.000	-4.345	1.39e-05 ***
RETURN	0.002736	0.152303	1.000	0.018	0.985667
ATHLETE	0.075154	0.520257	0.999	0.144	0.885140
FEMALE	-0.211158	0.093749	1.000	-2.252	0.024299 *
HSCA	-0.086218	0.079794	0.369	-1.080	0.279920
HSUS	0.586742	0.403044	0.912	1.456	0.145454
HSENG	0.421495	0.522758	0.866	0.806	0.420075
HSCHINA	-0.543726	0.179881	0.984	-3.023	0.002505 **
HSHK	-0.383966	0.249871	0.880	-1.537	0.124378
HSKOR	1.594.855	0.427698	0.989	3.729	0.000192 ***
HSOTH	-0.077980	0.243887	1.000	-0.320	0.749169
EOP	-0.012551	0.134346	1.000	-0.093	0.925570
NONCA	0.188861	0.672120	0.999	0.281	0.778716
FVISA	0.427938	0.610729	0.999	0.701	0.483490
year	-0.022395	0.021495	1.000	-1.042	0.297466
NRST	-0.892696	0.596481	1.000	-1.497	0.134496
MILITARY	5.072.556	1.651.552	0.960	3.071	0.002131 **

Table 63

	0	1
0	268	173
1	136	132

Finally we fit Islasso on the merged subset of academic and personal variables. We reported the results in Table 64. Comparing Table 53 and Table 64 we observe that once again Islasso reports fewer variables as significant compared to Lasso.

Table 64

	Estimate	Std. Error	Df	z value	Pr(> |z|)
(Intercept)	3.384.577	35.929.022	1.000	0.094	0.924949
PERMRES	-0.050459	0.056379	0.448	-0.895	0.370783
TAG	-0.013614	0.046195	0.508	-0.295	0.768219
HSUS	0.007196	0.037654	0.119	0.191	0.848437
HSENG	0.000040	0.327011	0.000	0.000	0.999902
HSCHINA	-0.382816	0.126027	0.980	-3.038	0.002385**
HSHK	-0.042358	0.050277	0.196	-0.843	0.399507
HSKOR	0.053499	0.097722	0.127	0.547	0.584063
HSOTH	-0.000014	0.029824	0.000	0.000	0.999635
year	0.002784	0.017863	0.884	0.156	0.876157
NRST	-0.165147	0.093181	0.851	-1.772	0.076339.
MILITARY	0.000030	0.212554	0.000	0.000	0.999886
MAJDUM	0.853150	0.234823	0.956	3.633	0.000280***
MINDUM	-0.234770	0.094189	0.968	-2.493	0.012683*
IGETCVER	-0.473260	0.094559	0.994	-5.005	5.59e-07***
met16A	-0.054181	0.057877	0.645	-0.936	0.349209
TUNITSUSED	-0.005030	0.007006	0.967	-0.718	0.472768
UNITSUSED	-0.025300	0.006805	1.000	-3.718	0.000201***
GPA	-1.200.021	0.162883	0.980	-7.367	1.74e-13***
STAYED	-0.914225	0.148530	0.983	-6.155	7.50e-10***
TOOKECN1B	0.000102	0.143655	0.000	0.001	0.999435
TOOKCMN3	-0.000023	0.035798	0.000	-0.001	0.999484
TOOKPLS21	-0.000060	0.181257	0.000	0.000	0.999737
TOOKECS15	-0.015346	0.047077	0.452	-0.326	0.744444
TOOKMGT11A	-0.069020	0.063759	0.652	-1.083	0.279028
TOOKMGT11B	-0.255452	0.087712	0.986	-2.912	0.003587**
TOOKARE18	0.000010	0.038757	0.001	0.000	0.999794
TOOKSTA13	-0.008376	0.038940	0.181	-0.215	0.829681
TOOKMAT16A	-0.135996	0.098522	0.612	-1.380	0.167475
TOOKMAT16B	-0.000017	0.206943	0.000	0.000	0.999933
TOOKMAT16C	-0.384001	0.088202	0.994	-4.354	1.34e-05***
APmacro	0.000008	0.021517	0.000	0.000	0.999696
Apcalc	-0.012060	0.044798	0.169	-0.269	0.787763
OSSJA	0.000059	0.223905	0.000	0.000	0.999789
Note:	*p<0.1; **p<0.05; ***p<0.01				

In Table 65 we computed the confusion matrix for Islasso and the misclassification rate we obtain almost 11.5% of false-negative predictions (type 2 errors) and 19.2% of false-positive predictions (type 1 errors). The misclassification rate is 30.85%, the lowest among all the model specifications.

Table 65

	0	1
0	323	137
1	82	168

In table 66 we reported the variables selected by each Lasso and the variables reported as significant from Islasso.

Table 66

X variables	Academic Lasso	Personal Lasso	Full Lasso	Academic Islasso	Personal Islasso	Full Islasso
Apcalc	x		x			
Apmacro	x		x			
Apmicro						
Apstat	x					
ATHLETE			x			
ENTRYMAJOR						
EOP			x			
FEMALE		x	x		x	
FVISA			x			
GPA	x		x	x		
HSCA						
HSCHINA		x	x		x	x
HSENG		x	x			
HSHK		x	x			
HSKOR		x	x		x	
HSOTH		x	x			
HSUS		x				
IGETCVER			x	x		x
MAJDUM	x		x	x		x
MET16A	x					
MILITARY		x	x		x	
MINDUM	x		x	x		x
NONCA						
NRST		x	x			x
OSSJA	x					
PERMRES		x	x		x	
RETURN						
STAYED	x		x	x		x
TAG		x			x	
TOOCMN3	x		x			
TOOKAMAT16C	x		x	x		x
TOOKAMT16A	x		x			
TOOKARE18	x		x			
TOOKCMN1			x			
TOOKECN1A						
TOOKECN1B	x		x			
TOOKECS15	x					
TOOKECS32A						
TOOKMATH16B						
TOOKMGT11A	x		x			
TOOKMGT11B	x		x	x		x
TOOKPLS21	x		x			
TOOKSTA13	x		x			
TUNITUSED			x			x
UNITUSED			x	x		x
year		x				

Moreover, in Table 67, we report Lasso AICs and Islasso AIC. We observe that Islasso results in lower AIC for every model specification.

Table 67

	Lasso AIC	**Islasso AIC**
Academic	3540.5	3487.4
Personal	3836.4	3825.9
Full	3503.4	3470.7

9 Conclusions

This thesis aimed to explore the application of linear shrinkage methods for variable selection to show whether these methods improve a model's predictive power. Moreover, we looked at ways to perform inference on the parameters selected via shrinkage.

Before starting with the novel dataset application, we explored how R and Stata worked, whether we could obtain matching results between the programs. To reach this purpose, we used the Mtcars dataset. We discovered that Stata and R adopt different scaling techniques when performing Ridge Regression. R's `glmnet` starts with a modified objective function, which is divided by the number of observations and has a standardized response variable. Therefore, to retrieve the original Ridge solution, we would need to scale the tuning parameter λ by a factor equal to $\frac{SD_Y}{N}$ and manually standardize the coefficients. To match Stata's `ridgereg` we would need to manually standardize the coefficients. On the other side, we found that obtaining matching results between R and Stata for Lasso regression was easier. For R's `glmnet` we would need to center y and standardize $\mathbf{X}$, while in Stata we need to remember to specify the option `penalized`.

Before starting with the analysis of the novel dataset, we analyzed different techniques to pick the tuning parameter. As highlighted in the theoretical section of this work (Section 3), selecting the tuning parameter λ is fundamental because it determines the degree of shrinkage used by Lasso and the objective function for any extension to Lasso.

We applied cross-validation (CV) to select a value for the parameter. It is common practice to use a $10 - folds$ cross-validation, but as we used LOOCV we decided to compare both techniques to highlight their differences. We found that λ_{min} selected by LOOCV typically falls in the middle range of the λ values selected with $10 - folds$. Indeed, LOOCV allows us to have no randomness in the folds and always obtain the same value of λ. We found that predicted values changed much according to the λ chosen to fit the model. This variability was much higher for Ridge than for Lasso estimated coefficients. Considering this variability, we believe that LOOCV is better because it ensures

replicability of results.

Once we understood how the program worked and how to pick the tuning parameter, we proceeded with an application of Lasso (and its extensions) to the EAWE dataset, to evaluate a model's quality of fit. We found that Lasso and every extension (Adaptive, Relaxed, and Islasso) improved the MSE compared to OLS. Specifically in Table 13, we observed that Adaptive Lasso with the parameter $\gamma = 0.1$ gave the lowest MSE across the different estimation methods. The adaptive Lasso is an extension where weights penalize different coefficients in the L_1 penalty.

Adaptive Lasso aims to make Lasso consistent in both selection and estimation. Therefore, these weights must come from unbiased and consistent estimates. Therefore, OLS is a natural choice, but according to the author, in the case of multicollinearity, Ridge regression can also be used to estimate the weights. Despite giving the lowest MSE, we are still concerned about the arbitrariness of this method. Therefore, we decided to use other strategies to obtain unbiased and consistent estimates for the application to our novel dataset.

Once this exploratory work concluded, we applied shrinkage estimators to the novel dataset composed of 126 variables and 4,091 observations of UC Davis transfers students. We aimed at understanding transfers students' performance, specifically why some students do not graduate within two years after the transfer.

Before estimating the model, we presumed that:

1. being a TAG student would decrease the probability of late graduation (TAG);
2. being an EOP student would increase the probability of late graduation (EOP);
3. a lack of solid math preparation would increase the probability of late graduation;
4. having two majors/a would increase the probability of late graduation (MAJDUM);
5. paying a non-residential student fee would decrease the probability of late graduation (PERMRES);
6. not having completed equivalent core courses at community college increases the probability of late graduation (IGETCVER)
7. switching majors increases the probability of late graduation (STAYED).

We divided the 126 variables into two subsets: academic and personal. We presumed that academic variables would be the most important in predicting students' late graduation. Therefore, to highlight predictive differences, we started by analyzing these subsets separately, and only as a last step, we merged the variables in a single design matrix.

The Lasso run with academic variables resulted in 31.59% misclassification error, 20.4% of type 1 errors and 11.1% of type 2 errors. In comparison, the Lasso run on personal variables results in a 44.01% misclassification error, 19.6% of type 1 errors and 24.4% of type 2 errors. Instead, when merging the variables, we obtained a misclassification error of 31.97%, 19.7 % of type 1 errors and

11.8% of type 2 errors.

Overall we conclude that academic variables are far more important for predicting late graduation as we obtain the lowest misclassification error and also the lowest AIC.

Finally, we applied Islasso. As explained in Section 2.3.2, Islasso aims to extend Lasso and obtain reliable p-values allowing hypothesis testing.

We first fit Islasso on the set of academic variables. MAJDUM, MINDUM, GPA, TOOKMAT16C, IGETCVER and STAYED were statistically significant. More specifically, having two majors increases the probability of late graduation (MAJDUM). Transfers students who completed the course requirement (IGETCVER), didn't switch majors (STAYED=1), had a minor (MINDUM), a higher GPA, and completed relatively more calculus classes (MAT16C) had a lower probability of graduating late. Overall, we obtained a misclassification rate of 31.31%, 20.2% of type 1 errors and 11.7% of type 2 errors.

Then, we fit Islasso with the set of personal variables. PERMRES, TAG, FEMALE, HSCHINA, HSKOR, MILITARY resulted statistically significant. Paying the non-resident tuition (PERMRES) being a TAG student (TAG), being a female (FEMALE), and coming from a Chinese high school (HSCHINA, i.e., a proxy for being Chinese) decreases the probability of late graduation. Instead coming from a Korean high school (HSKOR) and being a Military student (MILITARY) increases the probability of late graduation. Following the same pattern as Lasso, we obtained a higher misclassification rate than Lasso fit with academic variables. Specifically, we obtained a misclassification rate of 43.58%, 24.3% of type 1 errors and 19.1% of type 2 errors.

Finally, we ran Islasso on the merged variables matrix. The merged model reported HSCHINA, MAJDUM, MINDUM, IGETCVER, UNITSUSED, GPA, STAYED, TOOKMGT11B, TOOKMATC16C as statistically significant. As for the academic Islasso, having a double major increase the probability of graduating late. Instead, having a minor, having a high GPA, not switching between majors, having completed math and accounting classes (MAT16C and MGT11B) and a greater number of units transferred decrease the probability of late graduation. The only personal variable that resulted as statistically significant was HSCHINA, meaning that students who studied in a Chinese high school have a lower probability of late graduation. Islasso conducted on the entire set of variables gives us the lowest misclassification rate among all the model specifications with the lowest AIC and the lowest misclassification rate, 30.85%: 19.2% of type 1 errors and 11.5% of type 2 errors.

Overall, we believe that Lasso helped us understanding which variables belonged to the model and reinforced many prior presumptions on which variables should have entered in the model. Moreover, Islasso is a good compromise to close the gap between inference-selection as it allows us to perform variable selection and obtain reliable confidence intervals for a model's coefficients at the same time.

References

Athey, S. (2018). The impact of machine learning on economics. The economics of artificial intelligence: An agenda, 507-547.

Athey, S., & Imbens, G. W. (2019). Machine learning methods that economists should know about. Annual Review of Economics, 11, 685-725.

Beale, E. M. L., Kendall, M. G., & Mann, D. W. (1967). The discarding of variables in multivariate analysis. Biometrika, 54(3-4), 357-366.

Belloni, A., & Chernozhukov, V. (2013). Least squares after model selection in high-dimensional sparse models. Bernoulli, 19(2), 521-547.

Brown, B. M., & Wang, Y. G. (2005). Standard errors and covariance matrices for smoothed rank estimators. Biometrika, 92(1), 149-158.

Brown, W. G., & Beattie, B. R. (1975). Improving estimates of economic parameters by use of ridge regression with production function applications. American Journal of agricultural economics, 57(1), 21-32.

Chen, J. K. T., Valliant, R. L., & Elliott, M. R. (2019). Calibrating non-probability surveys to estimated control totals using LASSO, with an application to political polling. Journal of the Royal Statistical Society: Series C (Applied Statistics), 68(3), 657-681.

Cilluffo, G., Sottile, G., La Grutta, S., & Muggeo, V. M. (2020). The Induced Smoothed lasso: A practical framework for hypothesis testing in high dimensional regression. Statistical methods in medical research, 29(3), 765-777.

Dempster, A. P., Schatzoff, M., & Wermuth, N. (1977). A simulation study of alternatives to ordinary least squares. Journal of the American Statistical Association, 72(357), 77-91.

Dougherty, C. (2011). Introduction to econometrics. Oxford university press.

Efron, B., & Hastie, T. (2016). Computer age statistical inference (Vol. 5). Cambridge University Press.

Hastie, T., Narasimhan, B., & Tibshirani, R. (2021). The Relaxed Lasso.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning: data mining, inference, and prediction. Springer Science & Business Media.

Hastie, T., Tibshirani, R., & Wainwright, M. (2019). Statistical learning with sparsity: the lasso and generalizations. Chapman and Hall/CRC.

Hawkins, D. M. (2004). The problem of overfitting. Journal of chemical information and computer sciences, 44(1), 1-12.

Hocking, R. R., & Leslie, R. N. (1967). Selection of the best subset in regression analysis. Techno-

metrics, 9(4), 531-540.

Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: applications to nonorthogonal problems. Technometrics, 12(1), 69-82.

Hoerl, A. E., Kannard, R. W., & Baldwin, K. F. (1975). Ridge regression: some simulations. Communications in Statistics-Theory and Methods, 4(2), 105-123.

Huang, J., Ma, S., & Zhang, C. H. (2008). Adaptive Lasso for sparse high-dimensional regression models. Statistica Sinica, 1603-1618

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning (Vol. 112, p. 18). New York: springer.

JF, L. (1976). A simulation study of ridge and other regression estimators. Communications in Statistics-theory and Methods, 5(4), 307-323.

Jury, M., Smeding, A., Stephens, N. M., Nelson, J. E., Aelenei, C.,& Darnon, C. (2017). The experience of low-SES students in higher education: Psychological barriers to success and interventions to reduce social-class inequality. Journal of Social Issues, 73(1), 23-41.

McDonald, G. C., & Galarneau, D. I. (1975). A Monte Carlo evaluation of some ridge-type estimators. Journal of the American Statistical Association, 70(350), 407-416

Meinshausen, N. (2007). Relaxed lasso. Computational Statistics & Data Analysis, 52(1), 374-393.

Mullainathan, S., & Spiess, J. (2017). Machine learning: an applied econometric approach. Journal of Economic Perspectives, 31(2), 87-106.

Muniz, G., & Kibria, B. G. (2009). On some ridge regression estimators: An empirical comparisons. Communications in Statistics—Simulation and Computation®, 38(3), 621-630.

Sottile, G., Cilluffo, G., & Muggeo, V. M. (2019). The R package islasso: estimation and hypothesis testing in lasso regression. Technical Report on ResearchGate.

Taylor, J., & Tibshirani, R. J. (2015). Statistical learning and selective inference. Proceedings of the National Academy of Sciences, 112(25), 7629-7634.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B (Methodological), 58(1), 267-288.

Varian, H. R. (2014). Big data: New tricks for econometrics. Journal of Economic Perspectives, 28(2), 3-28.

Zou, H. (2006). The adaptive lasso and its oracle properties. Journal of the American statistical association, 101(476), 1418-1429.